TOWARDS AN EMBODIED MUSICAL MIND: GENERATIVE ALGORITHMS FOR ROBOTIC MUSICIANS

A Thesis
Presented to
The Academic Faculty

by

Mason Bretan

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Music

Georgia Institute of Technology
August 2017

TOWARDS AN EMBODIED MUSICAL MIND: GENERATIVE ALGORITHMS FOR ROBOTIC MUSICIANS

Approved by:

Gil Weinberg, Ph.D., Advisor
School of Music
Georgia Institute of Technology

Jason Freeman, D.M.A.
School of Music
Georgia Institute of Technology

Alexander Lerch, Ph.D.
School of Music
Georgia Institute of Technology

Guy Hoffman, Ph.D.
Department of Mechanical Engineering
Cornell University

Larry Heck, Ph.D.
Google Research
Google, Inc.

Date Approved: April 6, 2017

ACKNOWLEDGEMENTS

First, I would like to thank my advisor, Gil Weinberg, for introducing me to the field of robotic musicianship, providing the necessary guidance for me to develop this thesis, and fostering my passion for research through the continuous encouragement to explore the scientific and artistic opportunities our field has to offer. I am proud of everything we accomplished together over the years and I look forward to collaborations in the future.

My committee members have been incredibly helpful. Their feedback was integral to shaping this thesis and helping me to effectively and coherently present my ideas. Jason Freeman has been a supportive and important resource during this period. As a committee member and educator in general, he has gone above and beyond the call of duty with his diligence and mindful insights. Alexander Lerch's judicious comments, technical expertise, and rational thinking were essential for this research. Conversations about embodied cognition and robotics with Guy Hoffman were inspiring and helped to organize my thoughts about embodied musicianship. Larry Heck's supervision was invaluable and without his advice and guidance on machine learning this thesis would have likely turned out very differently.

I would like to thank Frank Clark, Chris Moore, Parag Chordia, Leslie Bennet, Joshua Smith, and Corissa James for providing support and advice during my time at Georgia Tech. I would also like to recognize Sageev Oore and Doug Eck who participated in helpful and stimulating discussions during my time at Google. I'd like to express my profound gratitude to my family who have been a never ending source of support and my girlfriend, Laura Hack, whose hard work and dedication to her own studies served as a source of inspiration for me to minimize my coding window, put

down my guitar, and stop looking for every excuse to procrastinate and finally write this thesis.

Finally, it is an honor to be the first to graduate with a Ph.D. in Music Technology from Georgia Tech. I thoroughly enjoyed my time as a graduate student in the School of Music and as a member of the Robotic Musicianship Group. Having the opportunity to conduct research in a fascinating field and perform in venues all over the world have provided me with rewarding and memorable experiences that I am sure I will cherish for the rest of my life. I look forward to seeing the Center for Music Technology and its degree programs continue to grow and flourish.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Embodied cognition is a theory stating that the processes and functions comprising the human mind are influenced by a person's physical body. The theory of *embodied musical cognition* holds that a person's body largely influences his or her musical experiences and actions. This work presents multiple frameworks for computer music generation as it pertains to robotic musicianship such that the musical decisions result from a joint optimization between the robot's physical constraints and musical knowledge. First, a generative framework based on hand-designed higher level musical concepts and the Viterbi beam search algorithm is described. The system allows for efficient and autonomous exploration on the relationship between music and physicality and the resulting music that is contingent on such a connection. It is evaluated objectively based on its ability to plan a series of sound actuating robotic movements (path planning) that minimize risk of collision, the number of dropped notes, spurious movements, and energy expenditure. Second, a method for developing higher level musical concepts (semantics) based on machine learning is presented. Using strategies based on neural networks and deep learning we show that it is possible to learn perceptually meaningful higher-level representations of music. These learned musical "embeddings" are applied to an autonomous music generation system that utilizes unit selection. The embeddings and generative system are evaluated based on objective ranking tasks and a subjective listening study. Third, the method for learning musical semantics is extended to a robot such that its embodiment becomes integral to the learning process. The resulting embeddings simultaneously encode information describing both important musical features and the robot's physical constraints.

CHAPTER I

INTRODUCTION

This thesis examines the computational operations that support automatic music generation by robotic musicians. Specifically, this work seeks to address the role of the physical body in the decision-making processes related to musical tasks such as composition and improvisation. Should the body merely serve as the interface connecting the cognitive and physical worlds? Or should the physical properties inherent to a robot determine the nature of the system's cognition? In this work the argument is made for the latter such that the generated ideas and physical actions result from a single integrated decision process. Compared to computational methods that are supported by disconnected modular processing pipelines, an integrative approach serves to identify solutions that are globally optimal across all facets related to generating and performing music. In this thesis several implementation techniques manifesting the idea of embodied processing are presented and evaluated.

1.1 Motivation

1.1.1 Unidirectional processing

Hurley describes the classical mainstream view of cognition as a "sandwich" in which thought (or cognition) serves as an interface between two completely separate and disconnected modules of perception and action [89]. This unidirectional theory of information processing and decision making can be used to describe the signal flow for generative music functions for most robotic musicians (**Figure 1**).

For example, a robotic musician typically employs an intelligence that supports a work flow as such:

1. Use traditional machine musicianship methodology to generate a note or sequence of notes (**Input** → **Musical Domain Knowledge** → **Music Output**)

2. Send the note(s) to a path planner (simulated action phase) that generates a sequence of movements that results in performing the previously generated note(s) (**Music Output** → **Simulated Action** → **Final Action**)

Figure 1: The typical signal flow of a robotic musician's decision processes mimics the unidirectional classical view of cognition and processing pipelines of many interactive music systems [169]. Given a stimulus, the system maps the input to a representation that it can be understood and analyzed (musical feature extraction and perception). The system then proceeds to make a decision about what note or notes to generate. Once this decision is made the system either makes further decisions about how best to physically actuate the notes given a set of physical constraints or directly sends the notes to physical actuators completely bypassing a simulated action stage.

On the surface this may seem sufficient, but consider the differences between software based machine musicianship and robotic musicianship – Software applications aren't bound to natural physics and as a result can be designed to play any note, combination of notes, timbre, volume, speed, and numerous other parameters. On the contrary, a robot operating in the physical world is limited in how it can move in space and, thus, limited by what it can sonically achieve. An optimal note sequence is not only one that is musically appropriate for the specific context, but also physically achievable given the constraints imposed by the system's embodiment. Therefore, the processing units that modulate music perception, reasoning, and performative action should have access to each other and provide continuous feedback in order to find these optimal solutions.

1.1.2 Embodied Processing

Though the sandwich theory of mental processing was considered mainstream 10-15 years ago, recently it is thought to be obsolete [137]. Instead, mounting evidence supports a portrayal of cognition and theory of mind that is decentralized and more cyclical in nature. This includes the *connectionist* approach in which behavior emerges from a combination of dynamical systems and interconnected networks of simple processing units [139]. This also includes the *embodied cognitive* approach in which mental reasoning and planning are shaped by features beyond the brain including the motor system and physical interactions with the environment [5]. These types of reasoning theories serve as the inspiration for the work in this thesis and provide general frameworks for how to incorporate a robotic system's physical body into the decisions that support its actions.

The term "embodied cognition" represents a diverse set of claims and hypotheses. The pervading theme of these claims suggests that the body plays a central role in shaping the mind. Wilson outlines six specific views to help disentangle the various viewpoints: (1) cognition is situated; (2) cognition is time-pressured; (3) we off-load cognitive work onto the environment; (4) the environment is part of the cognitive system; (5) cognition is for action; (6) offline cognition is body based [219]. This work primarily focuses on the last view and investigates how varying physical constraints of robotic systems can influence the perceptual and decision-making processes related to musicianship. Though there are various theories of mind and several examples of robots being used to study natural human behavior and brain science [7, 35, 6], it is not my intention in this thesis to prove or disprove a particular theory of cognition. Rather, I demonstrate why an algorithmic design inspired by body based cognitive processes is more suited for robotic musicianship than disembodied cognitivist approaches. I seek to demonstrate the benefits that an embodied reasoning system can have on generative music functions for robotic musicians and describe implementations that

can be used to glean these benefits.

To understand the nature of such benefits it is important to understand in the manner in which the previously described robotic musician processing work flow is deficient. The simulated action module consists of a process referred to as "path planning" that creates the movement plan necessary to perform the music generated by the preceding intelligence modules [118]. While it is possible that all of the notes are achievable by the robot, it is often the case that not all the notes can be played. In such a scenario a path planner may alter notes or drop notes completely, typically trying to devise a movement plan so that the most notes are played. This process is analogous to a composer writing a piece for a musician while not understanding the physical constraints of the human body or instrument.

Most composers write music capable of being performed by humans with two arms and two hands. Additionally, those studying composition learn what is and isn't humanly possible on different musical instruments [2]. Composers use their knowledge to make informed decisions addressing the physical world during the composition process. If people naturally had one arm and one hand these composers would write music suitable for such a physical form. They most likely wouldn't write music for two armed people and give the one armed performer the responsibility of finding a way to play it. For example, Maurice Ravel composed *Piano Concerto for the Left Hand in D major* for Austrian pianist, Paul Wittgenstein, who lost his right arm during World War I [84, 55, 213]. Wittgenstein was fully capable of performing the piece because Ravel addressed his specific set of physical constraints in compositional process. Likewise, the musical intelligence of a robotic musician should create music specific to its physical form that permits 100% of the notes to be performed all of the time. If the generated music is corrupted by the path planner then the intended output will not be realized.

It may seem reasonable if a path planner can regularly perform the vast majority of

the generated notes. A clever path planner may even include some of its own musical intelligence to make assumptions about what notes to drop or how to change them. One may argue that these dropped notes or note changes are a feature that results in interesting music indicative of robotic musicianship. Perhaps there is some truth to this as techniques employing randomness and chance have shown to be useful resources for artistic design [123, 93], however, the opportunity for interesting music to emerge as a result of a robotic musician's physical identity should not simply rely on the alterations made by a post hoc path planning operation. If robotic musicianship were to rely solely on these types of emergent behaviors then pieces like those made for Wittgenstein could only arise from chance. Furthermore, understanding the appropriate interactions between a physical agent and its instrument is an important aspect of musicianship [2]. Such knowledge should be addressed, not ignored. To progress robotic musicianship and explore the relationship between physicality and music it is important to design an intelligence that generates music not only *by* itself, but also *for* itself.

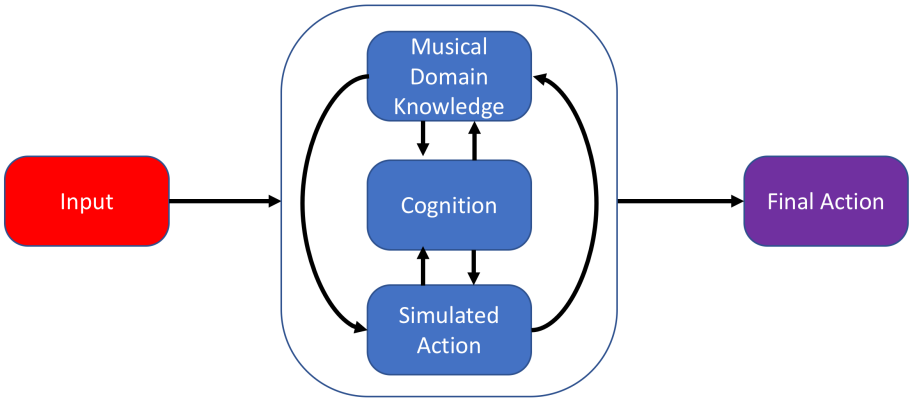


Figure 2: In the proposed approach to robotic musicianship, simulations of actions are performed and used to determine a global optimal musical output. Optimal in music may mean playing all the notes at the proper time or achieving some higher level musical semantic such as using pitches from a certain scale over a specific chord. Optimal physicality may mean solutions that reduce spurious movements, are energy efficient, and avoid collisions. This approach seeks to find a solution that jointly optimizes the musical and physical parameters.

I propose an approach to generative music in which embodiment and the physical

properties of a mechanical system help shape the musical decisions such that note sequences are jointly optimized for musical heuristics and physical parameters. Such a system would mimic the processes that are theorized to occur within embodied cognitive paradigms and perform continuous action simulation (**Figure 2**).

1.1.3 Musical Path Planning

Several advantages and opportunities emerge as a result of the physical embodiment of machine musicians compared to that of their pure software counterparts. These "robotic musicians" assume additional abilities to entertain, engage, socialize, and produce sound. Such desirable capacities are intrinsic to most social robotic platforms in general, as a result of physical presence and embodiment, and benefit both those directly immersed in the interaction as well as those simply witnessing it [111]. Though many of these interactive assets must be explicitly designed to address human perceptual and social tendencies, in music some advantageous characteristics arise entirely as a result of the inherent coupling between the robot's spatial movements and sound generation. The byproduct of sound generating movements is increased levels of rhythmic coordination and synchronization within ensembles because interacting musicians are able to anticipate the robot's musical onsets or behaviors through visual cues [82, 131].

These benefits, however, do not come without the additional constraints that are coupled to the natural world in which we live. An artificial intelligence (AI) controlling a mechanical body that is bound to the physical laws of nature must address its own presence in 3-dimensional space in order to function properly. This includes not only knowing the location of each of its degrees-of-freedom (DoFs), but also understanding how multiple DoFs need to behave and work together in order to complete a task [192, 109, 10]. Perhaps most importantly, the AI must also understand when a task is impossible (given its physicality) and anticipate failure before it causes damage to

itself or corrupts certain aspects of the task making success completely unattainable even with additional help.

In the musical domain these constraints are compounded by issues of timing. It is not enough to simply arrange the DoFs in a desired order; the timing and sequencing of the movements must also be considered. Often, depending on the time constraint and robot's physical design, the path in which each DoF moves and relocates itself must be optimized in order to reach specified locations in a timely manner. Path planning is the constraint satisfaction process of developing the proper coordinated movements.

In the domain of social robotics the constraints are even further compounded by human perception, as the physical behaviors of a robot influence how it is perceived. This is not a phenomenon unique to human perception of robots, but rather a consequence of the human tendency to personify robots and treat them as living things. Humans use physical cues to gauge a number of traits about other people or animals such as energy levels, emotional states, and competence. Humans attribute the same traits to robots [98, 202, 120]. For this reason it is argued path planning methods should be optimized not just for completing a task, but also for human perceptions and social constraints [184]. For example, there may be many possible movement sequence solutions enabling a robot to successfully carry out some undertaking, however, of these solutions there is likely a perceptual optimal that may eliminate spurious movements to avoid perceived confusion or may include secondary motions that are not essential to the task, but convey a message or adhere to some social construct.

In this work, I explore path planning and its relationship to the notion of *embodied musical cognition*, which states that an individual's body largely influences his or her understanding, experience, and decision processes pertaining to music [69, 121]. In particular, I examine how propioception and embodiment can (and I argue should) influence the musical decision processes and movement behaviors of an improvising

robot musician. The hypothesis is that a robot that utilizes a music generation method that jointly optimizes for its physical constraints as well as its general musical knowledge will more successfully convey its higher level musical ideas because the decisions will be informed by the capabilities of its sound generating motions. Additionally, such an integrated musical decision process will result in music that is defined by the robot's physical identity, hopefully leading to individualized styles.

Thus, the two primary benefits that yield from an embodied generative music system are:

1. **Optimality** – By integrating the musical heuristics with physical constraints a solution that is globally optimal across all parameters can be found. Though computing musical generative decisions and performing path planning as separate processes in series may produce reasonable music and prevent the machine from damaging itself in some cases, there are no guarantees that the higher level musical features or "musical semantics" will be maintained after path planning.

2. **Musical Emergence** – One of the goals of computer music is to exploit the behaviors of particular phenomena that can easily be computed by a computer, but not by a human as source material for generative art. Mathematical functions, algorithms, or various datasets may produce numerical sequences that can be mapped to sound. The resulting sonic output may have musical relevance or an inherent beauty that would otherwise not be discovered. Likewise, if the physical parameters of a non-humanoid robotic musician were to shape the system's understanding of music and generated musical decisions it is possible for non-human musical emergence to occur.

   A remarkable example of how a new set of physical constraints can lead to alternative viewpoints and new styles of music is epitomized in the story of legendary guitarist, Django Reinhardt. An 18 year old Reinhardt, already quite

musically accomplished, was left handicapped and disfigured from a fire. In his left hand his pinky and ring fingers were paralyzed leaving him only able to use his thumb, index, and middle fingers. He went on to devise a unique two-finger approach to playing that led to some of the most awe-inspiring and distinctive guitar-playing styles of the 20th century [53]. As a result of his handicap he was biased towards particular intervals and his style grew to include unique three-note chord voicings emphasizing 9th and minor-6th chords (a departure to what was considered normal practice at the time). Michel Des writes, "It is difficult to play standard scales with just index and middle fingers, so Django adopted an arpeggio-based rather than modal approach to soloing. He adapted arpeggios so that they could be played with two notes per string patterns which ran horizontally up and down the fret board instead of the usual vertical 'box' patterns, enabling him to move around the fret board with great speed and fluidity." Django Reinhardt is credited for developing "gypsy jazz" and his music has gone on to inspire other great guitarists such as Wes Montgomery [52].

1.2 Research Questions

There are several challenges and research questions embedded within the task of making an embodied generative music system. In this section, the primary research questions addressed in this thesis as well as the factors pertaining to each question are outlined. Integrating the physical and musical parameters has implications on the methods used for autonomous music generation, the possible representation of the system's physical states, planning methods, and on the application of machine learning when datasets are compiled from human performers with potentially very different physical constraints than the robotic musicians.

1.2.1 Autonomous Music Generation

a) **How can autonomous decision processes based on music incorporate the physical domains?** Including additional constraints on music requires a rethinking of the traditional machine musicianship concepts. Though previous methods for generating music are still relevant and can be useful resources, an integrated approach needs to address a significant expansion of parameters and complexity. Specifically, the note-level stochastic methods that are widely used may not be feasible for an integrated optimization method.

1.2.1.1 Deterministic Generation

There is no "wrong" or "right" method to autonomous music generation and numerous methods for programming a machine to generate music exist. Often the source material for the system's input has no immediate musical relevance (such as images, emotion plots, or earthquake data), but are mapped to control musical parameters. Other times natural phenomena resulting from mathematical equations (such as fractals or chaos) or stochastic sampling from statistical distributions are used to generate note sequences. The models that are merely audifying an input stream are considered translational models as they transform or translate the input from one form to another. Models that employ stochastic sampling are considered non-deterministic models because they possess a degree of randomness in the decision process.

For embodied musical decision-making the system must contain some measure capable of describing one note or note sequence as better or worse than another. Otherwise, the term "optimal" doesn't have any meaning. Therefore, in this research deterministic methods are used. At the most basic level a deterministic system considers multiple options, weighs them according to some metric, and makes a decision. The algorithm determines why one note or sample in a library is better than another note or sample. The operations that lead to the final decision can combine a

myriad of different variables that may be task or application specific.

1.2.1.2 Musicianship

Translational models or models that use randomness do not make decisions that are grounded in musical knowledge. In other words, these systems lack musicianship. Musicianship is traditionally thought of as the collection of knowledge and techniques an individual has to support music related skills. Rowe describes systems that employ musicianship as those that are capable of listening to music and making sense of what is heard, performing expressively, or composing convincing pieces [169].

There are many automatic music generation systems that lack musicianship, but are capable of creating interesting music [33, 119]. However, the premise for this research is that embodiment should influence the aspects of musicianship that are required for composing. This means that the processes for composing music such as the decisions about what notes to choose should be informed by the physical constraints of the system. Thus, the system should have the necessary degree of musical knowledge that enables it compose. Strategies to develop for this include rule and knowledge-based methods, grammars, and machine learning. In this work knowledge-based and machine learning methods are developed to encode musical knowledge and integrated with the physical properties of the robot.

1.2.2 Representation

In order to plan a path the space needs to be represented in a manner that the computer can understand. Is the space discrete or continuous? Is the size variable? Can it be represented as a grid and if so what resolution is necessary? Finding the best method to represent music is an open research question by itself without the constraint of embodiment. In this problem, however, the physical properties of a robot should not only be reflected in its musical output, but intertwined with its compositional decision processes. Therefore, the state space must simultaneously represent musical

and physical actions.

b) **What is the best way to represent the physicality of the robot within the decision process?** Some DoFs may need to be expressed individually (such as those directly involved in sound generation), but in order to reduce complexity other DoFs (such as those involved in generating sound accompanying gestures) need to be represented as a collective or as part of a higher level physical behavior. Designing meaningful physical behaviors that incorporate multiple DoFs, and various velocities, accelerations, positions, and trajectories will be essential.

c) **Is there a single integration approach that can be useful for many robotic platforms that have vastly different designs and functionalities?** A joint optimization methodology and single algorithm may be suitable for many music generating robots, however, a single state space representing all platforms is probably not possible. The physical design of robotic musicians tend to vary significantly from platform to platform. Instead, adjustments will likely need to be made that address a specific robot's physical characteristics and intended interactions and behaviors.

1.2.3 Planning

Once a map of the space has been established an algorithm is needed to search the space for an optimal solution. While many algorithms exist (A*, D*, Viterbi, Dijkstra's, Greedy) for performing this task their memory and computing constraints need to be considered. Additionally, the algorithm that is used will be the determining factor in how note sequences are generated. Understanding the consequences and advantages of any implementation is important for designing future musical applications such as autonomous composition, improvisation, and call and response.

d) **What aspects of the pure software generative music algorithms will need to be modified to better suit embodied methods?** While it might

be possible to create a completely real-time system in which the algorithm generates and plays notes on-line, such a method is probably not ideal. The nature of musical path planning is that an optimal sequence of moves is generated in order to achieve the musical goals. Therefore, rather than creating paths with a length of only a single note or move, the system should generate paths with lengths that represent complete musical ideas (though it is possible a complete idea is indeed only a single note). These musical chunks may be portions of phrases, complete phrases, or even entire structured improvisations from beginning to end.

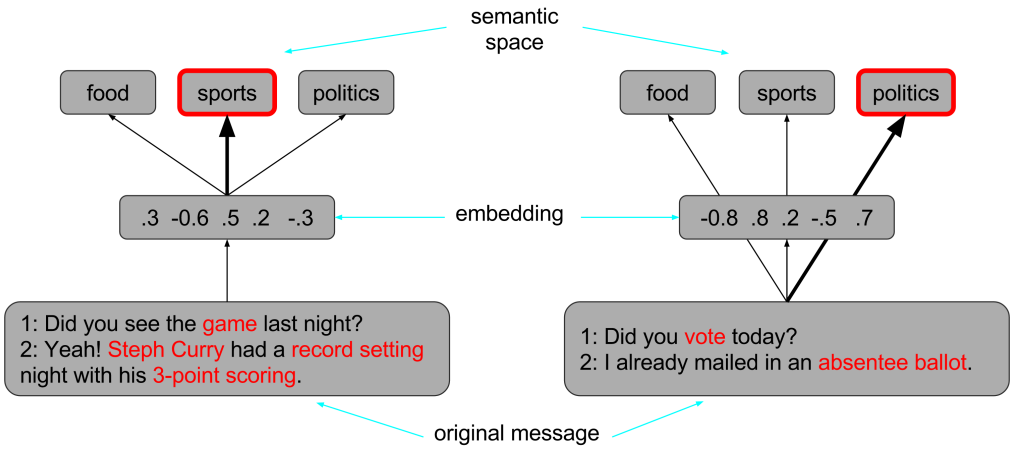


Figure 3: The original message is converted into a vector in the semantic space (a.k.a the embedding). While this vector still represents the original message, it is now doing so in terms of meaningful concepts instead of low-level ascii characters. Semantic embeddings allow us to describe the text or ask questions that have relevance to higher levels in the hierarchy of language. Instead of asking "do both of these sentences contain the letter c?" which we can do in a character-space, we can ask, "are both of these sentences about food?".

1.2.4 Learning

Machine learning and in particular deep learning has shown to be very useful for projecting low level representations into vector spaces that describe the data in terms of meaningful concepts. Projections into such a semantic space are referred to as

the semantic or latent "embedding." To understand this we can make an analogy to language (**Figure 3**). An effective embedding in language can describe higher level semantics that cannot be seen at the character level. For example, topics such as food, politics, or sports are revealed through phrases or sentences in language, not by calculating the probability that the character 'a' will follow 'r' or $P(a|r)$.

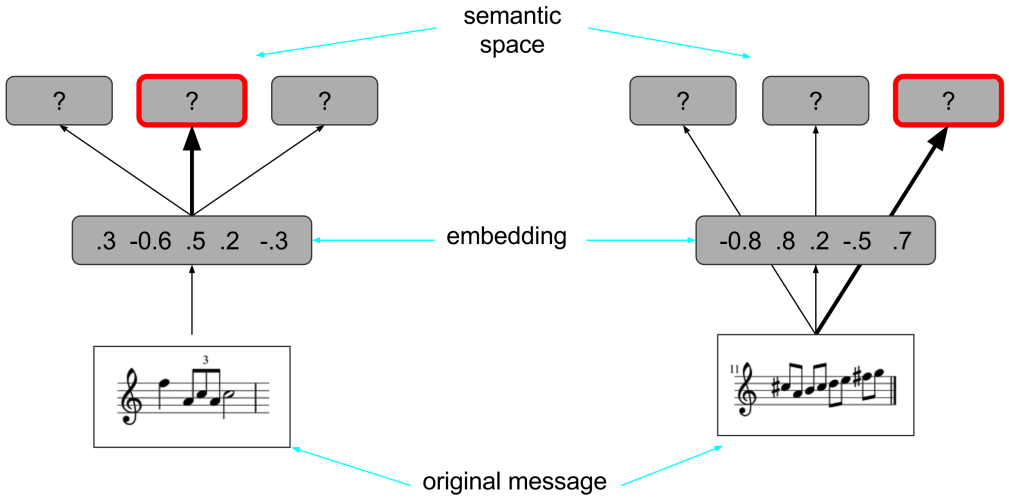


Figure 4: In the above figure the original message is music. Music, like language, has hierarchical characteristics and meaningful concepts describing higher-level features about a note sequence. It is necessary to learn what the appropriate meaningful concepts are in the first place and also how to map any given message into a vector relative to those concepts.

The idea of semantic representations can be applied to music as well. However, learning effective embeddings such that the learned semantics correlate with our own perceptions of music can be challenging (**Figure 4**).

e) **How can machine learning be used to learn an effective embedding space that captures meaningful musical semantics.** Meaningful vector spaces have been established by music theorists. For example, the "Circle of Fifths" and "Tonnetz" are geometrical projections that can be applied to pitches and chords [87]. Distances can be measured in these configurations and have relevance to harmony. Machine learning has the potential to learn similarly

effective spaces integrating various aspects of music.

f) **Is it possible to include aspects of embodiment in the learning process?** One problem of using machine learning to learn the musical semantics is that data is required. Given that the data would come from human composers and performers with physical parameters different than what a robot musician might have, the learning must try to generalize high level concepts in such a way that they are still useful for robots of various shapes, forms, and physical constraints.

g) **If an effective embedding space is learned how can it be applied to musical tasks such as composition or interactive call and response?** Establishing an effective space is extremely valuable because it can provide a system with musicianship on both perceptual and generative levels. As such, applications such as call and response can be established by examining the information encoded within the embedding space.

1.3 Contributions

A commonly held view representing the relationship between machine musicianship and robotic musicianship is that the former encompasses all research related to machine generated music and is a component of robotic musicianship. Historically, machine musicianship has referred to software applications [169] and as such the generative methods designed for robotic musicians have been interchangeable with pure software musicians. The algorithms tend to be constructed to merely generate notes (or sounds) that are musically significant. Though machine musicianship and robotic musicianship are undoubtedly related and share many of the same objectives and challenges, physical embodiment is such a polarizing and distinctive attribute that new methods for optimally incorporating a sense of physical self into an algorithm should be developed. These methods would be unique to robotic musicianship. If machine

musicianship sets out to algorithmically find a sequence of notes that is musically significant, robotic musicianship sets out to find a sequence of notes that is musically significant, physically possible, and its musical character can be enhanced through simultaneous physical behaviors.

The three main traits that are used to highlight the significance of embodiment and to differentiate robotic musicianship from machine musicianship include 1) acoustic sound production, 2) natural physics-based visual cues from sound producing movements, and 3) expressive and communicative sound accompanying movements. While the study of these differences and algorithmic development for machines with such traits have led to unique and note-worthy performances, I propose that a focus on only these three traits is insufficient for encompassing all of the interesting facets that result from combining software musicians with mechatronics. Therefore, I introduce a fourth trait to address the relationship between embodiment and musicianship, *embodied musical cognition*. The work in this thesis makes several contributions by describing algorithmic designs for addressing this trait and demonstrating the value of the trait using several objective and qualitative evaluative metrics. This includes:

1. A path planning algorithm for optimizing movements to play pre-composed note sequences [Chapter 3].

2. A generative algorithm integrating physical properties with knowledge-based musical heuristics [Chapter 3].

3. A method for developing a musical embedding space using deep learning [Chapter 4].

4. A subjective listening study to evaluate the learned space [Chapter 4].

5. A generative method integrating path planning with the learned semantics [Chapter 4].

6. A model that learns to generate note sequences optimized for a system's specific physical parameters [Chapter 5].

CHAPTER II

EMBODIED MUSICAL COGNITION AND RELATED WORK

The theoretical foundations for the various designs, methods, and experiments presented in this work are derived from fields of computer music, music theory, cognitive science, artificial intelligence, and machine learning. While *robotic musicianship* is the pervading theme of this work, additional concepts are addressed including algorithmic composition, embodied cognition, path planning, and machine learning.

2.1 Embodied Musical Cognition

2.1.1 Theory and Evidence

The theory of embodied musical cognition is a model that illustrates the processes of the human musical mind. Unlike the dualist theory of *cognitivism*, which treats the mind as a distinct entity independent from the body, this theory states that the manner in which a person perceives, generates, and interacts with music in all domains is dependent on the properties of that person's body [121]. The cognitivist approach has been used to explain various structures in music such as tonal tension [125, 127], melodic peaks [60], and melodic similarity [79]. Though these models are successful in explaining musical structure for a large body of music, because they are reliant on an implementation of systematic rules they fail to explain structure for musical genres that do not adhere to the rules such as microtonal, serialism, or free jazz [16, 44, 171]. These models are considered to represent human musical cognition in a narrow sense [220].

Recently there has been mounting evidence that the musical mind is influenced

by the motor system. Phillips-Silver and Trainor explored the interactions between the auditory and vestibular (regulation and perception of movement and balance) systems. In a study, published in Science Magazine in 2005, they demonstrated that movement influences the auditory encoding of rhythmic patterns in human infants. In the study they found that infants preferred to listen to the rhythm that matched their particular metrical movement form [160]. In a follow up study in 2007 with adults, they had similar results suggesting that there is a cross-modal interaction between body movement and auditory encoding [161]. As such, they declared that humans really do "feel the beat."

Physical action has also been shown to influence visual perception [14, 9], however, in perceptual mechanisms that require sub-second processing this bias is either reduced or absent. Iordanescu et al. found that action enhances auditory sub-second temporal sensitivity, but not visual sub-second temporal sensitivity [90]. Their research was inspired by the fact that people naturally dance to music and rhythmic auditory stimuli help to facilitate precisely timed physical body movements. In the study, the reciprocal of this phenomenon was tested. Using a temporal-bisection task, participants listened to a sequence of three brief clicks or watched three flashes (all within 507ms). They were tasked with determining whether the second click or flash was closer to the first or third. Some participants listened or watched passively while others initiated the clicks/flashes with a keyboard press. An increase of auditory temporal precision was shown when participants actively initiated the clicks, but not when actively initiating the flashes. In a similar study, Manning and Schutz found that moving to the beat in music improves temporal perception [138]. They show that by tapping while listening to music, listeners were more likely to identify the correct tempo after a period of silence when the music ended.

As result of our embodiment, humans develop associations between physical actions and certain effects (known as action-effect associations). In 2011 Sedlmeier

et al. performed a study to examine the impact of real or imagined body movements during music listening [173]. Specifically, participants were asked to listen to music while either activating or inhibiting positively associated muscle groups (such as those controlling smiling). The study found that participants had higher preference for the music they listened to while activating the positively-linked muscle groups. Similarly, Maes and Leman further show that expressive action patterns can influence the perception of music. Children's perception of music was shown to modulate under different scenarios of performing either happy or sad dance choreographies to emotionally ambiguous music.

These works are part of a growing body of evidence and literature demonstrating a sensorimotor integration that influences musical cognition. While these studies focus on the perceptual aspects of music and do not demonstrate modulations on the compositional or performative characteristics of music, there is anecdotal evidence the physical parameters of an individual influence decisions in the compositional and improvisation realms (described in the next section). Furthermore, if cognition is not unidirectional and instead comprised of a perceptual $\Longleftrightarrow$ action feedback loop then such modulations would inherently exist.

2.1.2 Associative Learning and Embodied Processing in Musicians

According to the theory of embodied musical cognition thinking and acting are intertwined and complex cognitive processes result from the joint process. Based on this theory, a person that simply studies music theory without playing an instrument experiences music differently than an instrumentalist; both may have a profound musical knowledge, but the instrumentalist's physical interaction with music has molded his or her musical mind making the experience different. There is a hypothesis for explaining this "molding" phenomena – associative learning processes promote internal models representing sensory-motor relationships [205, 21]. In other words, the experience and

witnessing of repeated specific action-effect scenarios create perceptual and expectation priors in the brain. Indeed, there is empirical evidence that demonstrates musicians have developed certain biases that make them respond differently than non-musicians.

In 2005 Haslinger et al. compared the effects of observing finger movements related to piano play and non-piano play between expert pianists and naive musical controls [78]. Participants observed the movements in an fMRI machine. The results demonstrated a significantly stronger activation in brain areas associated with mirroring of the pianists compared to the control participants. Mirroring is the behavior in which a person subconsciously imitates the behaviors of another individual. Many believe that this helps one to predict or understand the intentions of others [112, 166, 167, 67].

In 2009 Repp and Knoblich examined the effect of learned associations of action on auditory perception [165]. Participants played pairs of octave-ambiguous tones by pressing successive notes on either a piano keyboard or computer keyboard. They were asked to identify if the interval within an octave pair ascended or descended. They found that pianists gave significantly more ascending responses when the order of the key presses was left-to-right. These results suggest acquired action-effect associations can influence auditory cognition.

Additional qualitative and anecdotal evidence suggests that some of the higher level musical semantics that describe a person's style emerge as a result of that person's physical interaction with his or her environment. It's not just that a musician has learned to play an instrument, but the type of instrument and specific physical interactions the body has with that instrument can encourage the musician to develop very particular action-effect associations. Gibson (2006) explores instrument specific musical tendencies in jazz improvisation and based on qualitative interviews with professional musicians speculates that some patterns or motifs partially arise due to the natural affordances of the instrument on which they're being performed [68]. For example, the trumpet's nature makes it more suitable for small intervals going

up and down a scale as opposed to arpeggios, something that the piano is almost perfectly designed for and may be the reason why arpeggios are so prevalent in pianists' improvisations. One musician describes Eric Dolphy's playing style on saxophone containing massive intervals as "bloody impossible on trumpet." Another musician explains that there are more notes available in a single hand position on electric bass than double bass and as a result one tends to play more four note patterns on electric and more three note patterns on double bass.

The affordances of an instrument are born out of the combination of its physical design and a person's ability to interact with it given his or her own physical limitations. Given the above comments, addressing the affordances of the physical instrument seems very important, but autonomous generative music methods used by robotic musicians do not integrate this element in the decision processes. Understanding one's instrument and the associated action-effects built into it is extremely important when an individual is learning and this understanding is considered to be a significant aspect of musicianship [94, 22].

Instruments for robots can be designed in any number of ways and, thus, the physical parameters of each design has artistic potential assuming the intelligence functions using a higher level reasoning system that integrates both the cognitive and the physical. Yet, understanding the parameters of an instrument is only the first step. An embodied intelligence also must understand the physical parameters of its own body. A combination of propioception, situated awareness, and understanding of physical limitations is necessary to develop optimal solutions.

In the above examples, associations are learned from a musician's ongoing and repeated situatedness with his or her instrument. It can be argued that being human and merely existing with a very specific set of physical limitations has influenced the way we perform and compose music. In a 2002 article in the journal *Music Perception* Vijay Iyer suggests that embodiment and the physical constraints of the

human body effects performative aspects of music [91]. He examines microtiming in African American groove music (such as jazz and afro-cuban). He hypothesizes that microtiming variations serve not only to highlight the structural aspects of musical material and to fulfill some aesthetic function, but also reflect specific temporal constraints imposed by physical embodiment. Thus, the microtemporal variations seen in human performance are not just random deviations due to human imperfections, but are a result of our physical constraints. In one example, he suggests that the rhythmic technique of spreading ("tripletizing" duple rhythms) results from the physical limitation of human beings being only able to tap (with a hand or finger) up to seven taps per second and microtiming deviations stem from this. He goes on to deconstruct a Thelonious Monk solo and hypothesizes that many of the decisions Monk made were a result of the current physical constraints imposed by a combination of the musical scenario and the current position of Monk's body. For example, pianists tend to choose keys that lie under the current hand position over keys that do not.

Though Iyer's hypotheses cannot be readily proven or disproven, he suggests that because there is a degree of regularity in the microtiming variations seen in groove music (they're not random) it is likely that the regularity stems from some universal characteristic (beyond aesthetic preference) shared by everyone such as the physicality of the human body. Beyond his research, Iyer's comments from the perspective of an extraordinary musician are also quite insightful. One quote is especially pertinent to the work of this thesis:

> A skilled improvisor is always attuned to the constraints imposed by the musical moment. This requires an awareness of the palette of musical acts available in general, and particularly of the dynamically evolving subset of this palette that is *physically* possible at any given moment [91].

2.2 Robotics

2.2.1 Embodied Intelligence in Robotics

The centralized cognition paradigms that employ a unidirectional processing flow pipelined through *separate* modules emulating perception, thought, and action are commonly used in artificial intelligence applications, however, they are typically seen in disembodied software applications. Feedback, cyclic processing, and integrated motion planning becomes more relevant when an agent is given a body and situated in a physical environment. In fact, the entire field of control theory is the study of dynamical systems in which behavior is continuously modified by feedback. In robotics the feedback comes from a combination of motors (a sort of propioception and sensory motor system) and situatedness. However, this feedback controls very low level parameters and typically does not address the higher level decisions a system would need to make in order to successfully navigate complex tasks such as interacting with people.

A large emphasis of robotics research over the years has focused on developing methods that enable a machine to effectively adapt to an environment that is dynamic and unstable while armed with only a basic knowledge about its own physicality [135, 190, 148]. Brooks introduced this type of model in 1991 [30]. He described an agent that has no higher level symbolic representation of the world and its overall intelligence emerges from a set of independent low level behaviors. Though in these examples the robots' environments are typically tangible and materialistic, robots designed for musical applications function under similar conditions, as music is also characterized by its dynamic and changing nature. The challenges in music are, therefore, similar and include tasks such as quantifying constraint parameters, developing useful fitness functions, and efficiently finding optimal trajectories.

Though the notion of integrating cognitive and physical parameters into a unified intelligence for music generation and performance is relatively new, there are examples

of such a unification in other high level robotic applications. For a robot to learn to complete different tasks involving its environment it naturally needs to apply a sense of physical self. In one system described by Shadmehr and Mussa-Ivaldi, a robot learned to adapt its motion primitives (such as reaching) to different forces applied to its motors, while still effectively achieving the task [174]. Though there is an ideal or independent kinematic plan for the motion primitive, the system is able to adapt its physical trajectories and electrical current draw based on dynamical forces. The musical analogue to this process would be when a robot has a general musical plan, but based on information regarding any physical limitation, finds a solution that addresses the constraints while still allowing the general musical plan to be achieved. For example, if the higher level goal is to play a phrase that goes up in pitch then the robot would find a sequence of notes that is both playable and ascends in pitch.

Similarly, a constrained optimization approach has been described by Kapoor and Taylor for assistive surgical robotics in which a multi-robot system jointly optimizes for physical constraints and task objectives [99]. The task objective is defined as the input provided by the surgeon and the constraints are defined from the motors' and individual robots' Cartesian coordinates. A distributed algorithm is used so that there are several smaller optimization problems computed for each individual robot and a final optimization problem that is solved encompassing all the robots. This distributed methodology allows the solution to be solved quickly and enables real-time control for the surgeon. Such a methodology can be useful in the musical domain as well in which multiple robots are used. For example, a robotic ensemble that autonomously choreographs a dance to music would take the musical signal as the task input (where the music is mapped to physical behaviors) and each individual robot would find a local dance move that is appropriate or congruous with the ensemble as a whole.

Efficiency and dimensionality reduction is one of the pervading arguments for embodied architectures. Bicchi et al. regard the human hand as a *cognitive organ* and

its nature determines behavior, skills, and cognitive functions [17]. The design and analysis of robotic hands can be simplified by applying the principles of how the brain efficiently leverages all of the physical characteristics of the hand. Gabbiccini et al. demonstrate this by creating optimal grasping forces with an artificial hand using a reduced kinematic space often referred to as *postural synergies* or the *eigengrasp space* [66]. This subsequently led to more natural and enhanced prosthesis control [162].

Embodied processing methods are also used to navigate a robot's interactions with people. Hoffman describes a computational framework of embodied cognition for autonomous interactive robots [81]. The implementation is based on three principles: (1) modal perceptual representation; (2) action-perception and action-cognition integration; (3) a simulation-based model of top-down perceptual biasing. The perception, cognition, and action modules are intertwined allowing for what the agent learns from observing the human to be directly applied to its own actions. This type of processing improved the robots‘ efficiency and fluency of its interactions and resulted in faster reaction times.

It has been shown that complex and adaptive behaviors can arise in an artificial agent simply as a result of including the agent's physical body as a component of the mental functioning and reasoning process [6]. This seems to be a better solution than explicitly designing every rule addressing how the robot must function with its environment as per the cognitivist approach, especially considering the complex and chaotic nature of the physical world. In fact, there has been a recent trend to include embodied cognition in robotic systems [225, 145, 23, 198]. Embodied cognitive methods also have relevance to musical applications which have similar complexities to the problems addressed in these examples.

There is mounting evidence that the mind is indeed embodied [122, 63, 193, 195]. Regardless of whether this is true, it seems not only logical, but necessary for a robot to have an embodied cognition. The alternative to integration would be to

create individual and separate modules for each function or variable (music knowledge, physicality, social gestures, etc.) However, in creating separate modules we risk failing to encapsulate the higher order dynamical system that contextualizes each function and illustrates the reason for their ontogenesis in the first place. This idea is the premise for 'developmental robotics' in which a robot's higher level processes emerge as a result of its physical interaction with the environment [6, 215]. A robot's physicality bridges the gap between its internal cognition and the physical infrastructure of the environment by providing it with the ability to gain information essential to autonomous learning and decision making [117]. In order for this ability to be realized the robot must have an understanding of how to interact with the environment given the constraints defined by its physical embodiment.

2.2.2 Robotic Musicianship

Researchers of robotic musicianship seek to develop robots capable of using a musical instrument for performance in both solo and interactive scenarios. In order to achieve this, the robots must have the physical means for actuating sound as well as the necessary underlying intelligence to support automatic music generation and interaction. Thus, robotic musicianship is typically understood as the intersection of 1) *musical mechatronics* or the study, design, and manufacture of mechanical systems capable of generating sound acoustically and 2) *machine musicianship* or the study and development of algorithms pertaining to aspects of musical intelligence such as perception, performance, composition, and theory. Traditionally, research in machine musicianship has focused on software applications that respond to and generate music [130, 56, 216].

While researchers encounter challenging and interesting questions from a purely scientific and technological perspective, there is also impetus for pursuing this field of study that stems from cultural and artistic ambitions. Robert Rowe elegantly

argues that, "if computers interact with people in a musically meaningful way, that experience will bolster and extend the musicianship already fostered by traditional forms of music education (...) and expand human musical culture" [169]. Though Rowe was referring to pure software applications, robotic musicianship is motivated by similar artistic pursuits, yet with the added feature of embodiment. In [29] we argue that the ultimate goal of robotic musicianship is to supplement and enrich the human musical experience by exploring the opportunities and challenges that arise from giving a machine musician a mobile physical body such as the visual cues, acoustic sound, and social interactions.

2.2.3 Robotic Musicianship – Design

There is no single physical design of a musical robot that is perfect and there have been no claims that one such design exists. Instead, designers make trade-offs that may account for a robot's size, mass, possible anthropomorphic design, the instrument it will play, the specific genre(s) it will play, method of sound actuation, ability to provide useful visual cues, the ability to provide social cues, energy consumption, price, and aesthetics. In other words, designers make decisions that are influenced by both music-specific ambitions and physical characteristics.

There are several examples of musical mechatronics with vastly different designs that allow performance on different musical instruments. The Logo's Foundation has developed several non-anthropomorphic percussive, string, and wind playing robots [136, 164] that are midi-controlled. Similarly, Expressive machines Musical Instruments (EMMI) and Karmetik have developed different drum and string playing robots with impressive mechanical control [168, 103, 104]. At Georgia Tech, the Robotic Musicianship Group works with the percussive robot, Shimon, that uses an anthropomorphic design [150]. Shimon is additionally able to provide social cues using its head gestures [24]. Each of these robotic systems exhibits varying degrees of

autonomy and intelligence. However, each similarly possesses a musical intelligence system that is independent of its physical control parameters.

The tables below (Tables 1, 2, 3, and 4) offer brief descriptions of many different types of robotic systems used in music. Typically, the term 'robotic musician' is used to emphasize that the system contains functions and skills related to musicianship. A robotic musician is distinguished from a 'musical robot' in which the primary focus is the design and mechatronics [210, 29]. The systems listed in these tables emphasize the breadth of physical designs and applications. Additionally, each is classified as being either a robotic musician or musical robot to highlight the primary research focus of the system. For a complete survey with more in depth descriptions of various robotic musicians, design incentives, and functionalities see [29].

Table 1: Robots that play percussive instruments.

Developer	Description	Type
Logo's Foundation	**Vibi** - An automated vibraphone with individual dampers for each bar.	Robotic Musician
	Troms - Seven single drums each outfitted with a a set of strikers positioned at various locations on the drumhead.	Robotic Musician
	Vacca - Series of cowbells equipped with different style hammers driven by solenoids [136, 164].	Robotic Musician
Expressive Machines Musical Instruments (EMMI)	**MADI** - *Multi-mallet automatic drumming instrument* consists of 15 solenoid strikers positioned around a single snare drum allowing the system to take advantage of different timbres available at specific locations on the drum [168].	Robotic Musician

Table 1 Continued

Trimpin	**Automated idiophones** - Pitched percussion outfitted with individual solenoids that drive strikers that can be interchanged and adapted to achieve different timbres [199].	Musical Robot
Karmetik	**Notomotion** - A drum equipped with several rotary and pull solenoids with a rotating mounting structure allowing for a wide range of timbres.	Robotic Musician
	Raina - A rainstick attached to a drive train that slowly rotates [102].	Robotic Musician
Eric Singer	**LEMUR bots** - Solenoid based striking mechanisms designed to play individual drums [183]	Musical Robot
Georgia Tech (GTCMT)	**Haile** - Anthropomorphized robotic percussionist equipped with a linear motor and solenoid for striking a drum [210].	Robotic Musician
MIT	**Cog** Robot - Robotic drummer using smooth oscillators for controlling rhythmic motion [218].	Robotic Musician
Mitsuo Kawato	**Humanoid drummer** - A humanoid drumming robot used to study human drumming mechanics [7].	Musical Robot
University of Washington	**Piano playing hand** - Anatomically correct piano playing hand designed to study human kinematics [224].	Musical Robot

Table 2: Robots that play string instruments.

Developer	Description	Type
Logo's Foundation	**Hurdy** - An automated bowed bass intstrument that uses several motors to bow the strings.	Robotic Musician
	Aeio - An automated cello in which the strings are excited by using two electromagnets driven by a two phase signal on opposite sides of the string [136, 164].	Robotic Musician
EMMI	**PAM** - PAM: Poly-tangent Automatic multi-Monochord [168].	Robotic Musician
Trimpin	**Krantkontrol** - An installation of 12 guitar-like instruments with a plucking mechanism and solenoids for fretting.	Musical Robot
	"If VI was IX" - An installation containing hundreds of guitars with pitch and plucking actuators [199].	Musical Robot
Baginsky	**Aglaopheme** - A slide guitar with solenoids for plucking and fretting and a motor used to alter the position of the bridge [8].	Musical Robot
Victoria University of Wellington and New Zealand School of Music	**MechBass** - A four string modular robotic bass guitar player. Each string is plucked using a stepper motor and a solenoid attached to a carriage moves along the string via a belt drive for fretting [142].	Musical Robot
Eric Singer	**Guitarbot** - Wheel (embedded with picks) capable of rotating at variable strings to pluck guitar strings. The bridge rides on pulleys driven by a DC motor. [183].	Musical Robot

Table 2 Continued

Sergi Jorda	**Afasia** - Solenoid are used to pluck string and push solenoids press the strings on the bridge [97].	Musical Robot

Table 3: Robots that play wind instruments.

Developer	Description	Type
Logo's Foundation	**Ob** - An automated oboe in which the reed is replaced with a acoustic impedance converter that models a real reed in a human mouth cavity [136, 164].	Robotic Musician
National ICT Australia	**Robot Clarinet** - A robotic clarinet player that controls for blowing pressure, lip force, and lip position to affect pitch, sound level, and spectrum [4].	Musical Robot
Roger Dannenberg	**McBlare** - A bagpipe playing robot that uses an air compressor and electro-magnetic fingers for pressing the sound holes [49].	Robotic Musician
Waseda University	**Sax and Flute Robot** - Robot designed with a focus on controlling finger dexterity, lip and tongue control, and lung control (via an airpump) [188].	Robotic Musician

Table 4: Augmented robotic instruments.

Developer	Descritpion	Type
Logo's Foundation	**Dripper** - A rain machine which controls the precise size and frequency of each drip [136, 164].	Robotic Musician

Table 4 Continued

Augmented Instruments Lab (Queen Mary University)	**Magnetic Resonator Piano** - An acoustic grand piano augmented with electromagnets inside the instrument to create vibrations in the strings [141].	Musical Robot
	Digital Bagpipe - An electronic bagpipe chanter interface that is equipped with infrared sensing to record finger positions for later analysis [143].	Musical Robot
Karmetik	**ESitar** - A hyperinstrument that uses sensors to measure human gesture during sitar performance and interface directly with a computer [103].	Robotic Musician
Trimpin	**Contraption Instant Prepared Piano** - A prepared piano system capable of bowing, plucking, and vibrating piano strings [199].	Musical Robot

2.2.4 Robotic Musicianship - Physical Influence and Embodied Intelligence

The research in this thesis primarily focuses on the machine musicianship aspect of robotic musicianship, however, design is undoubtedly important and must be addressed in the system's intelligence if the music making decisions of a generative algorithm are to be influenced by the robot's physicality. I argue that similarly to the processes that go into formulating the physical design of a musical robot, a robotic musician should have an intelligence that integrates the cognitive and physical domains such that the musical and physical behaviors are a result of a decision process that jointly optimizes musical goals, path planning, and human perception.

There are many examples of robotic musicians using the bodies of interacting humans to influence the system's musical outputs. Pan et al. use computer vision

enabling their humanoid marimba player to detect the head nods of an interacting musician [158]. Solis' anthropomorphic flutist robot similarly uses vision to detect the presence of a musical partner and once detected the robot then listens and evaluates its partner's playing [185]. Using a Microsoft Kinect, a person uses his arms to control the specific motifs played by the Shimon robot and the dynamics and tempos by moving his arms in three dimensions [24]. Additionally, a percussionist is able to train the system with specific gestures and use these gestures to cue different sections of a precomposed piece [24]. Mizumoto and Lim incorporate vision techniques to detect gestures for improved beat detection and synchronization [131, 147]. In their system a robotic theremin player follows a score and synchronizes with an ensemble by using a combination of auditory and visual cues.

These examples, however, are not representative of embodied intelligence. Instead they follow the classical unidirectional flow of sensor input $\rightarrow$ musical decision $\rightarrow$ action. Examples of embodied intelligence are those that utilize a feedback system to update their physical parameters or undergo a simulated action phase to inform musical decisions.

In one example of a dynamical system, Kapur et al. developed a robotic harmonium called *Kritaanjli*, which extracts information from an interacting human performer's style of harmonium pumping and attempts to emulate it [104]. The robot's motors use information from a linear displacement sensor that measures the human's pump displacement. A Pololu DC motor is used to pump the harmonium and is equipped with a hall-effect encoder, which provides feedback to modulate the pump action. This system uses body sensors (motor encoders) to regulate and adjust its behavior. There is no simulated action, but there exists active physical feedback.

Similarly, Jo et al. recently (2016) introduced a violin playing robotic system that employs auditory and contact force feedback enabling the robot to modulate its sound in real-time [95]. The system listens to the sound it produces on the instrument and

modifies the bow's contact with the strings and velocity of the bowing gesture in order to establish good tone.

In a drumming application design for the MIT *Cog* robot, the system uses dynamic control to create smooth arm movements. Each joint is equipped with a dedicated local motor controller and the microcontroller generates a virtual spring behavior at 1kHz, based on torque feedback from strain gauges in the joints [217, 218]. Similarly, the GTCMT drumming prosthesis uses a proportional-derivative (PD) controller to modulate the behaviors of two DC motors. The modulations influence the spring or "bounciness" of the stick in order to make buzz rolls or double bounce strokes [25].

These systems are examples of using embodied feedback to control very low-level attributes of musicianship. They influence behaviors describing how to play, but not necessarily what to play.

Composers incorporate their own knowledge of the robots' bodies and create compositions or applications that specifically address the unique physical designs. While there is no artificial intelligence, the physical uniqueness of the systems are leveraged by hardcoding or manually manipulating them to behave in particular manners.

In *Bafana*, a composition by Gil Weinberg, the Shimon robot plays various pre-composed motifs. Shimon's unique physicality is leveraged by hardcoding it to play multiple motifs simultaneously with its eight arms [24]. In the piece *Skies* by Govinda Ram Pingali, I composed a part for the robotic drumming prosthesis. It takes advantage of the two sticks attached to the prosthesis and the fast rate at which they can play. Jason Barnes also composed a piece, *Earth*, using similar techniques to take advantage of the unique abilities of the prosthesis.

Recall that one objective of algorithmic composition is to re-purpose the inherent behaviors and characteristics of computational functions such as chaos, genetic, and cellular automata for a musical context. The algorithms may lack any meaningful

musicianship, yet, yield interesting behaviors that can be useful source material for generating music. In an installation by Trimpin, the acoustic behavior resulting from the interaction of sounds within the physical world was leveraged by distributing hundreds of automated guitar systems within a room [199]. This was achieved through manual manipulation and configuration of each instrument and not through an automated embodied intelligence.

An example of algorithmically leveraging interesting behaviors in the physical world was shown by Albin et al. in a swarming robotics system that generated music based on different swarming behaviors [3]. The robots' absolute physical positions and relative positions to one another were used for musical mappings. Though physical parameters were used to generate music, the specific parameters did not come from the robots' own physicality and designs, but rather their locations, therefore, nullifying the possibility for interesting music to occur that is algorithmically born out of physical identity. Additionally, Trimpin and Albin's systems are closer to translational or sonification models in which the musical output is modulated by the behavior of interesting algorithms and designs, but do not incorporate computational musicianship.

Perhaps the most relevant example of embodied influenced music generation was done by Hoffman et al. [83]. In this work physical gestures (movement sequences) of the Shimon robot were modeled instead of note sequences. The gestures were the building blocks such that the music emerged from the nature of these physical movements, rather than the traditional approach of using symbolic representations of music to drive the movements. The work in this thesis describes a scenario in which the music and physical movements are intertwined within a decision process that simultaneously joins both elements.

2.2.5 The utility of coherent sound accompanying movements in music performance

There is also motivation for a robot to obtain a physical-musical cognition that goes beyond the constraints and opportunities related to sound producing movements. Humans often employ multi-modal communication processes when interacting with one another. A musician's secondary movements, or sound accompanying movements, are not only useful in making a performance more entertaining and engaging, but also serve more functional purposes by providing the ability to communicate intent and influence how observers interpret the music.

In one study, Vines et al. show that a musician's movements serve to both augment and reduce the perception of tension in music [201]. It was also found, in this study and several others, that observers use the visual cues from movement to help understand the performer's internal states, concepts of phrasing, and changes in emotional content [203, 48, 132]. The differences between seeing a musical performance and simply listening to one are characterized by observers' perceptual and even physiological responses [32]. One experience may not be better or worse, but the differences exist and a robotic musician should leverage the relationship between ancillary motion and musical features to enhance its ability to communicate its own musical goals and interpretations.

Understanding how physical behaviors are connected to human perception is important for effective application of a robot's sound accompanying movements. Nussek and Wanderley found that the multi-modal experience from watching a human perform is more dependent on the overall movement characteristics of the whole body and relative motion of limbs to each other rather than specific arm or torso movements [154]. This was also found to be true in a study with a small 5-DoF faceless robot in which the overall physical motion characteristics (such as velocity, acceleration, and periodicity) were much more effective for communicating sentiment compared to

individual DoF positions [27].

Additionally, entrainment, or the process of two independent rhythmic sequences synchronizing with each other, is an essential characteristic of the physical movements that accompany music and should be addressed by a robotic musician's movement generation processes. Entrainment not only helps to synchronize and coordinate with other interacting musicians, but is also useful in communicating both broad and immediate temporal order and structure within the music [39].

CHAPTER III

SIMULATED ACTION AS A PLANNING TASK

Sections from this chapter have been prepared and published in:

> **Bretan, Mason and Gil Weinberg. "Integrating the Cognitive with the Physical: Path Planning for an Improvising Robot." AAAI, 2017.**

In this chapter a proof of concept demonstrating the utility of an embodied musical cognition for robotic musicianship is described. The problem is approached as a path planning task and uses search techniques to develop solutions that jointly optimize for the robot's physical constraints and musical semantics. The necessity for planning is demonstrated in an experiment that evaluates the planning algorithm against a baseline algorithm which makes locally optimal decisions (across physical constraints and musical semantics) in a greedy fashion. This experiment addresses the optimality goal of embodied processing. In the second part of the chapter the goal of musical emergence is addressed in the context of jazz improvisation. A knowledge-based implementation of jazz is described and the resulting generative musical path planning system is capable of creating different musical motifs in which the source of variance stems from the physical constraints. This allows for efficient and autonomous exploration of the relationship between music and physicality and the resulting music that is contingent on such a connection. The system is qualitatively examined and applied to the Shimon robot and used in performance settings.

3.1 Introduction

Computational tasks that attempt to find optimal solutions or sequences are a hallmark of artificial intelligence. Often what makes the task difficult or interesting is computational intractability. NP-complete puzzles require clever methods to prune pathways and nodes in order to make the problem solvable (or at least capable of finding sufficient local solutions). A classic example is that of chess playing computers. Of course, a chess game can be considered solvable because every possible sequence of moves in every possible game amounts to a finite number. However, this finite number is so large that a 'brute-force' method of a complete game is not feasible. Instead, a brute-force method of the next n possible moves is computed with emission scores that rely on metrics evaluating the strength and potential of different pieces and their respective locations, as opposed to simply relying on whether the move lies on a path that leads to winning [149]. The pathways and next moves are then chosen using different search algorithms (including minimax, A*, iterative deepening, depth-first search, and alpha-beta pruning) that jointly optimize for many parameters or metrics.

A similar method is used in this work. However, in order to function properly in a musical context the appropriate state space and useful metrics must be established. Just as the number of possible games of chess is insurmountably huge, the number of note sequences that can be composed is equally massive. Moreover, a pianist can choose to play a middle 'C' with any one of his or her ten fingers; including physical parameters expands the possible options, thus, the decision-making process becomes more expensive. Therefore, good solutions must be identified without a brute-force search over all possible state sequences. The following sections describe a strategy for achieving this.

3.2 Musical Path Planning

As a first step towards demonstrating the effects of integrating the cognitive and physical in music generation a general path planning method is needed. Ignoring the goal of musical emergence and excluding generation for the moment, the first task is to develop a strategy so that a robot can perform a precomposed set of notes. Not only should the robot perform these notes, but it should do so in manner such that it avoids physical damage to itself, energy is conserved, and the perceptual advantages of visual cues (for people witnessing the performance) are not lost (i.e. avoid spurious movements). The following traits, listed in order of importance (for this system), are considered:

1. **Physical Harm** – Avoid catastrophic movements such as limb collision.
2. **Music Quality** – Maximize the number of notes it plays in the precomposed sequence.
3. **Perception** – Avoid spurious movements to maintain effective sound-producing visual cues.
4. **Efficiency** – Minimize distance traveled, torque applied to the motors, or power consumption.

3.2.1 Musical C-Space

In order to make optimal decisions that address the above factors a map representing the planning environment is needed, also known as the configuration space or *C-space*. In this case, the environment consists of possible musical notes as well as the physical configurations of the robot. Though music can be thought of as a continuous space in terms of intonation and timing, typically, it is represented as a discrete space. The notes on a musical score are represented by a finite number of possibilities of pitch and duration and mapped out according to the their absolute temporal position (again

finite) within the context of the score. Even dynamics and tempo are discretized in score representations (fortissimo, mezzo forte, piano, presto, andante, largo, etc.) and it is the conductor or performer's interpretation that serves as a function to project these cues into a continuous space. Here, music is represented discretely as individual notes and in an attempt to minimize complexity only pitch and rhythm are addressed. Musical features including dynamics, articulation, and temporal expression (rubato, accelerando, etc.) are not represented in the map.

Similarly, the poses and movements of a robot can be represented continuously, however, the physicality is reduced to a discrete space in order to integrate with the musical representation. Discretizing the C-space is highly typical in search and path planning methods for robots. The discrete space is usually represented using Voronoi diagrams, regular grids/occupancy grids, generalized cones, quad-tree, and vertex graphs [170]. A single state in this space may describe a posture, orientation, or even a motion primitive depending on the task. In this work the manner in which the space is discretized is motivated by the objective of the robot's ability to play a sequence of notes. Therefore, a single physical state should somehow be integrated with the available musical states.

In this implementation, a single state represents the pitch or pitches that can be reached given a specific physical configuration. The complete C-Space describes all possible configurations of the motors that can be used in order to play all possible pitches. As an example the design and specific physical constraints of Shimon, a four armed marimba playing robot, is considered [150]. An example of a state is shown in **Figure 5**. Velocity is not addressed in the C-Space and duration is addressed using a transition function between states (described more in the next section). Additionally, Shimon is a percussionist so the duration of a single strike is constant, though the resting space between notes is variable allowing for rhythm. Such a constraint is analogous to a trumpet or clarinet player using only staccato articulated notes.

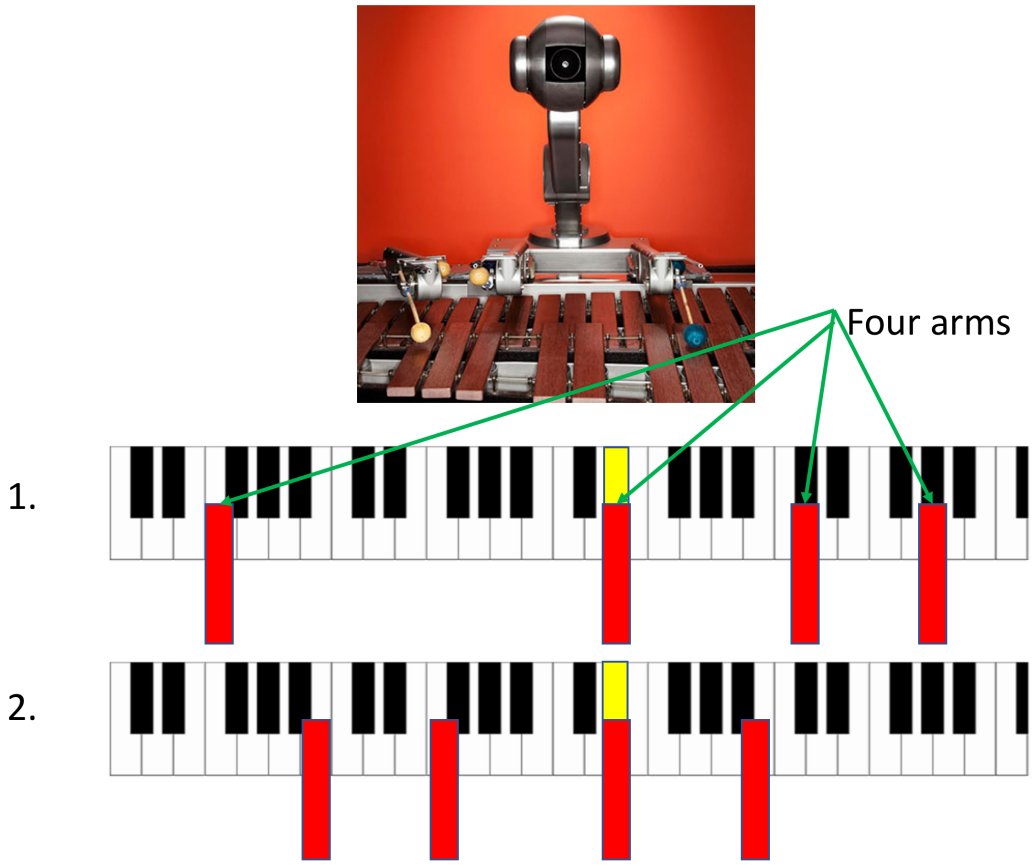


Figure 5: Shimon has four arms which can be configured in many possible ways. A single state in the C-Space describes the possible physical configurations of the arms capable of playing a specific note(s). In this figure two states in which Shimon can play the highlighted 'Eb' are shown.

3.2.2 Planning

Now that the state space has been established a method for choosing particular states is needed (**Figure 6**). While the designs of many robotic musicians prevent them from damaging themselves, Shimon's four arms share the same physical space making propioception integral for damage prevention. Alternative designs in which a solenoid is permanently stationed over every key do not pose such a risk. However, for any system in which the moving parts share the same space a form of path planning is necessary. Shimon's design limits its ability to play certain musical passages. For example, because Shimon's arms move on only one axis some transitions at certain

speeds are impossible such as pitch intervals less than a minor third with temporal intervals less than 130ms. This is due to two factors: (1) the speed at which a single arm can transition from one bar to another (130ms for a half step) and (2) the width of Shimon's arms is larger than a single marimba bar meaning two arms can't previously be positioned to play a minor second very quickly. Defining the physical constraints that impose these limitations in a manner a computer can understand is essential.

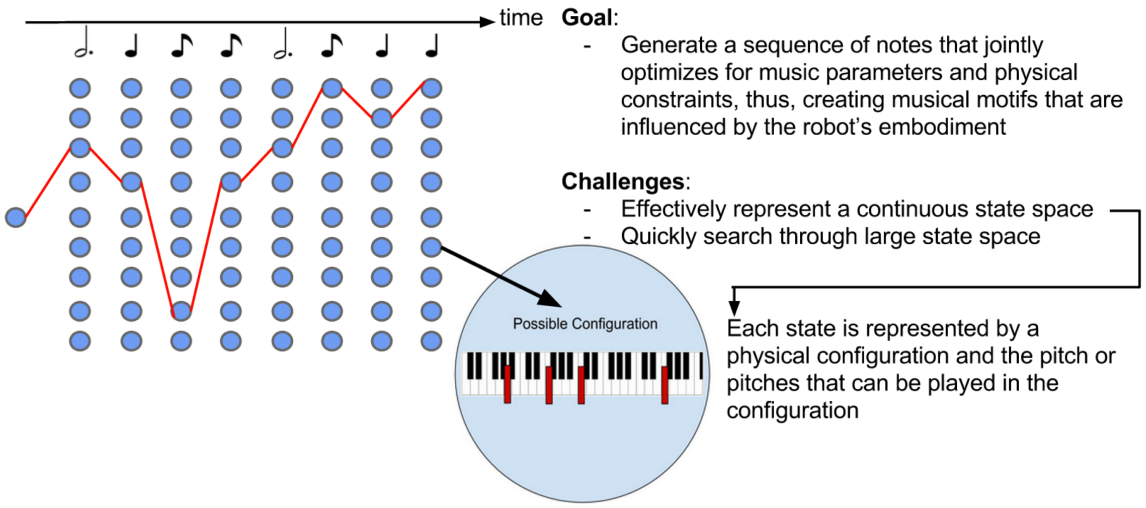


Figure 6: General premise for finding a path through states in the configuration space.

The cost of each possible state is derived from the attributes enumerated above (physical harm, music quality, perception, and efficiency). It is computed using a combination of each state's instantaneous (or emission) score as well as a transition score describing the movement from one state to another. The variables involved include the state space, S, observation space, O, sequence of observations, Y, transition matrix A, and emission matrix B. When playing a precomposed set of notes the observation space is made up of all the possible pitch and rhythm combinations. The composition then represents the sequence of states within this space. Transition and emission matrices are typically defined as probability matrices that describe the likelihood of certain events occurring. In path planning these matrices do not describe

probabilities, but rather binary descriptors of whether an event is possible with added heuristics to encourage specific types of behavior.

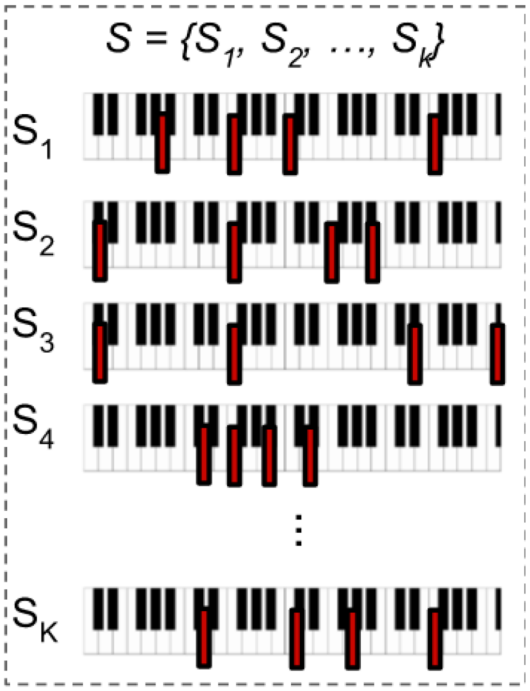
S is the state space $S = \{S_1, S_2, S_3, ..., S_k\}$, where k is the total number of states and S_i denotes the ith state in S (see **Figure 7**). Recall, that a state represents a physical configuration of Shimon's arms. The emission score of a given state, B_{S_i}, is described by its ability to play the pitch in note(s), Y_n, where Y_n is the nth set of notes to be played in the observation sequence.

$$B_{S_i} = \begin{cases} \alpha & Y_n \in S_i \\ 0 & Y_n \notin S_i \end{cases} \tag{1}$$

where α is a weight describing the strength of the parameter. For playing precomposed note sequences there is only one instantaneous parameter, which is the musical heuristic or *musical quality* parameter. In this scenario the value simply depicts whether the pitch of a given note can be reached by the physical configuration represented in the current state. The weight, α, and subsequent weights are determined empirically on a held out test set.

A state only describes a static physical configuration and the possible pitches that can be reached from that configuration. Therefore, to consider attributes pertaining to time (such as note intervals) a measure describing the transition between states is necessary. Though this is computed as a matrix A, this matrix is not static, but tempo and interval dependent and thus must be recomputed for different tempi. Therefore, the transition matrix, A, can also be thought of as the result of a function that evaluates the quality of transitions between states. The quality is determined according to the physical harm, perceptual, and efficiency heuristics described below.

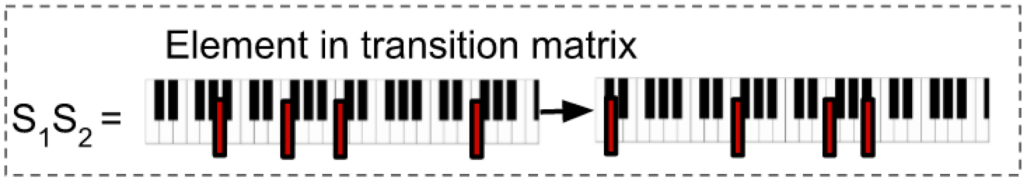
Physically possible transitions, as allowable by the design of the robot, can be determined a priori and formalized in a matrix, P. This matrix can be thought of as a mask of zeros and ones where a value of one denotes a possible transition and zero an impossible transition. Here, a penalty is imposed on the cost for attempting impossible



Each state, *Si*, defines a physical configuration of the robot's four arms. Certain notes can be played depending on the configuration.

The emission matrix, *B,* is defined by the possible notes playable in each state.

Figure 7: State representation and emission score.



The transition matrix defines transitions between configurations. A tempo dependent mask is applied to the matrix allowing only physically possible. transitions

Figure 8: The transition between two states.

transitions. The physical harm heuristic, h, describing the transition between state, S_i, and state, S_j, is computed as:

$$h_{S_i,S_j} = \begin{cases} 0 & for \quad P_{S_i,S_j} = 1 \\ -\beta & for \quad P_{S_i,S_j} = 0 \end{cases} \tag{2}$$

where β is a weight describing the strength of the parameter. Though this metric serves to penalize and prevent harmful movements, in practice, however, it is better to use a method that *guarantees* prevention of collision. This can be done by pruning the

impossible states from the search and selection process, thus, preventing the system from ever making a bad move.

A significant argument for robotics in music is the presence of visual cues associated with sound producing movements which allows interacting musicians to anticipate what is the robot is going to play. For example, when a pianist moves his or her hand to a particular position over the piano, the viewer can reasonably assume that he or she is going to play in the pitch range defined by that location. The perceptual heuristic is an attempt to minimize the number of times a movement associated with sound production does not produce sound. At times, this may be necessary to avoid collision. In Shimon, for example, movement of multiple arms may be necessary to play one note (recall that the width of Shimon's arms are bigger than the width of a marimba key). Finding paths that reduce instances of this will help preserve a person's ability to utilize the gestural cues and visually anticipate what will be played. In a perceptually optimal transition the cardinality of the set describing the difference between the transitioning sets, $S_i \rightarrow S_j$, should be equal to the cardinality of set, O_n. Therefore, the perceptual heuristic, p, between two states can be defined as:

$$p_{S_i,S_j} = \begin{cases} 0 & for \quad |S_j - S_i| = |Y_n| \\ -\lambda & for \quad |S_j - S_i| > |Y_n| \end{cases} \tag{3}$$

where λ is a weight describing the strength of the parameter.

Efficiency should be considered when there is a measurable amount of energy associated with a movement. This is largely defined by the robotic system and instrument design. For Shimon, the distance each arm travels is directly correlated with the amount of power drawn for each motor of its motors. However, for a device like the robotic drumming prosthesis the values of the PD controller may predict power draw [59]. The efficiency heuristic between two transitioning states is just defined as a variable, d_{S_i,S_j}. For Shimon, specifically, it is defined as:

$$d_{S_i,S_j} = -\omega \sum_{n=1}^{4} |\vec{S_{i_n}} - \vec{S_{j_n}}| \tag{4}$$

in which the distance traveled (in terms of pitches) is summed across all four arms and ω is a weight describing the strength of the parameter.

The final transition between two states is then defined as:

$$A_{S_i,S_j} = h_{S_i,S_j} + p_{S_i,S_j} + d_{S_i,S_j} \tag{5}$$

And the total cost, C, of transitioning from $S_i \rightarrow S_j$ is defined as:

$$C_{S_i,S_j} = B_{S_j} + A_{S_i,S_j} \tag{6}$$

The proper weights for each parameter are needed in order for the system to perform with the desired behavior. In this implementation preventing physical harm has the highest priority followed by playing the most notes, preventing spurious movements, and finally movement efficiency. To achieve this prioritization the weights must be such that $\beta > \alpha > \lambda > \omega$. However, in the actual implementation β is not actually used because because transitions that would cause a collision are simply removed from the state space. The other weights were manually assigned based on empirical testing with $\alpha = 1.0$, $\lambda = .5$, and $\omega = .1$

3.2.2.1 Greedy Selection

As a baseline method, a greedy search method is implemented. In a greedy search method the cost, C, is used to make selections regarding movements deterministically for every single note. Once the selection has been made the robot makes the necessary movements to play the note and moves on to the next note. The only information that is considered is the current configuration state and the possible next states to which it can transition. With Shimon, one more musical heuristic can be added – the possibility to play notes in alternative octaves. When a certain note cannot be reached then the note's pitch can be re-evaluated in different octaves. The octave

transposition results in note sequences that more closely resemble the rhythm of the original sequence, but may not be desirable and does not need to be performed. The greedy search algorithm is described in pseudocode below.

Algorithm 1 Greedy Selection

```
1: function GREEDY(S, current, y) : X
2:
3: // Given the current state evaluate all next possible states
4: // for the observation y
5:     for each state i from 1 : size(S) do
6:         V[i] ← Cost(current, S_i, y)
         X = arg max(V)                          ▷ The best state.
7:     return X
```

3.2.2.2 Viterbi Search

The baseline does not find a global optimum given a sequence of notes, but rather makes greedy decisions at the rate of an individual note. In Shimon, the behavioral result is that when a note is received the arm closest to the note is chosen to move and strike the note. This may seem reasonable, however, in practice many moves are not possible because of speed limits and the width of each arm being greater than the width of a bar on the marimba (meaning the note cannot be reached without collision). To prevent collision between arms a note can be transposed to a different octave and the method is repeated. In more extreme cases in which no arm can play the original or a transposed version of the note then the note is dropped completely.

A more optimal solution can be found if the system has access to future notes. In this case, the algorithm can plan ahead so that an immediate move helps to set the robot up for success for playing subsequent notes. If the physical limitations of a robotic system are restrictive then planning for multiple notes at a time can be useful for creating a more suitable movement sequence.

Given the state space representation there are two suitable algorithms that lend themselves to this type of planning. The first is the A* (A-star) algorithm. A* is a

pathfinding search algorithm that is guaranteed to find a solution if one exists. The algorithm uses heuristics to guide its search, however, it is necessary for the heuristic to be admissible. This means that the heuristic never overestimates the true minimum cost of reaching the goal. In music there is no clear admissible heuristic and without the heuristic A* is essentially breadth-first search. Instead, this work uses a beamed Viterbi search algorithm to generate an optimal sequence. Other than no admissible heuristic, the major benefit of Viterbi is that the computation time is stable. In musical applications timing is very important, therefore, having this stability can inform the developer how to best design various applications or interactions with the robot.

In order to decode the graph using Viterbi then the state space, S, observation space, O, sequence of observations, Y, transition matrix A, and emission matrix B are necessary. The algorithm traverses the states in the graph to generate multiple hypotheses that maximize the given parameters. These parameters are identical to the ones outlined above and used in the greedy selection process. The pseudocode for the implementation is written below and the general concept for finding a good sequence is provided in **Figure 9**.

Considering that the total number of arm configurations is about 73,815 (each arm is capable of playing roughly 36 notes), finding the global optimal path is not always feasible if the path is to be found in a timely manner. Therefore, a beam is applied in order to prune unreasonable branches quickly. The beam narrows the search to only look at states capable of playing the observed note in the sequence, Y. This significantly reduces the number of possible states and allows the system to compute paths for precomposed note sequences quickly enough for interactive applications.

Algorithm 2 Viterbi Path Planning

function ViterbiDecode($S, Y, A, B, start$) : X

// Initialize scores with starting state and first note in observation sequence
    for each state **i** from $0 : size(S)$ **do**
        $V[1, i] \leftarrow B_{start,i} \cdot \max_k(V[start, k] \cdot A_{k,start})$
        $P[1, i] \leftarrow \arg\max_k(V[start, k] \cdot A_{k,start})$ ▷ Store back-pointers

// Compute scores over entire state lattice
    for each time step **i** from : $size(Y)$ **do** ▷ Iterate through observations
        for each state **j** from $0 : size(S)$ **do** ▷ Iterate through each state
            $V[i, j] \leftarrow B_{ji} \cdot \max_k(V[i-1, k] \cdot A_{kj})$
            $P[i, j] \leftarrow \arg\max_k(V[i-1, k] \cdot A_{kj})$

// Trace the back pointers beginning from the best end state
// in order to identify the optimal state sequence
    $z_{size(Y)} \leftarrow \arg\max_k(V[i-1, size(O)])$ ▷ Identify best end state
    $x_{size(Y)} \leftarrow state_{z_size(Y)}$
    for each time step **i** from $size(Y) : 2$ **do** ▷ Trace backpointers
        $z_{i-1} \leftarrow P[z_i, i]$
        $x_{i-1} \leftarrow state_{z_{i-1}}$
    return X ▷ The optimal state sequence.

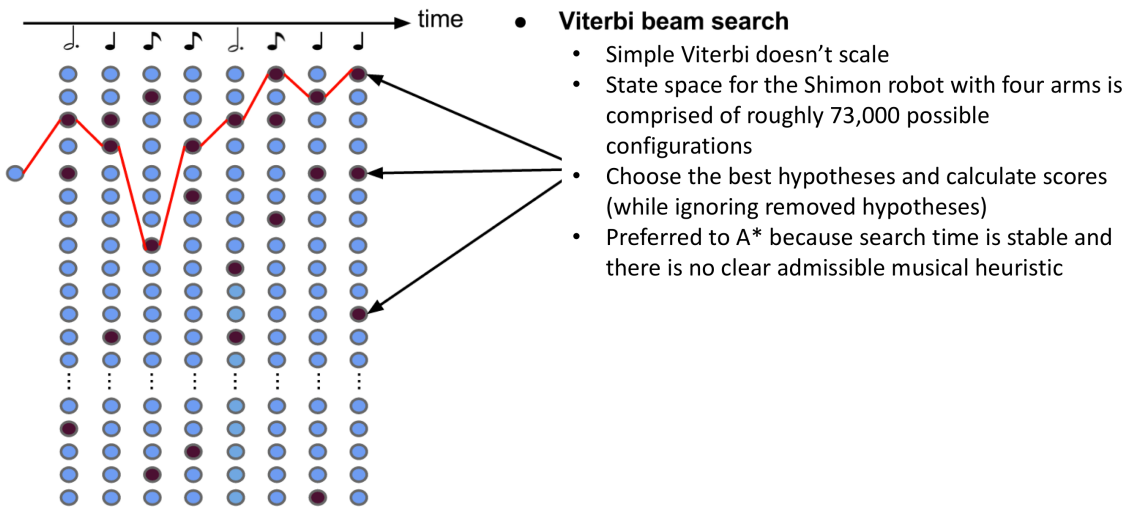


Figure 9: A beamed Viterbi search is used to traverse the state space and find good movement sequences as defined by the heuristics.

3.2.3 Evaluation

The effects of simulated action and planning can be measured objectively by comparing the Viterbi based metric to the greedy selection process. In the first experiment both algorithms were given one hundred pre-composed monophonic note sequences. The sequences came from a mix of publicly available classical melodies and transcribed jazz improvisations with high degrees of technical complexity such as Andersen's 18 Etudes for flute and Coltrane's improvisation on Giant Steps. The make of the dataset is shown in Table 5.

Using the two algorithms, note sequences, using the physical constraint parameters of Shimon, were generated. The rate at which each algorithm transposed a note and dropped a note was computed. Additionally, the average distance traveled over four beats of music was also computed (across all arms). The unit of measurement for distance is the width of a marimba bar, therefore, a distance traveled equal to one is equivalent to one arm moving by one note. A distance traveled equal to four can be equivalent to one arm moving four notes, two arms moving two notes each, four arms moving one note each, and so on. In order to remove a zero distance bias for when a

Table 5: Distribution of data in test set.

Performer/Composer	Number of Pieces	Original Instrument
Carl Stamitz	5	Clarinet
Carl Maria Weber	6	Clarinet
Joachim Andersen	18	Flute
Ernesto Kohler	12	Flute
Dan Adler	10	Guitar
John Coltrane	6	Saxophone
Dexter Gordon	8	Saxophone
Sonny Rollins	7	Saxophone
Sonny Stitt	14	Saxophone
Clifford Brown	5	Trumpet
Freddie Hubbard	9	Trumpet

Table 6: Greedy vs Planned experiment 1 results.

	octave transpose rate	drop rate	normalized distance
greedy	39.2%	14.9%	2.8
Viterbi	11.1%	1.7%	3.1

note is dropped the distance is normalized by the number of notes played. The results are shown in Table 6.

The experiment was repeated, however, a set of physical constraints were used for a hypothetical robot. The new constraints were similar to those of Shimon, but instead the simulated system is equipped with two arms with each having a width equal to one half of one marimba bar (meaning it is possible for the two arms to play a minor second interval without moving). With this set of constraints there are no dropped or modified notes as it is possible with both the greedy and planning algorithm to play all notes in the monophonic sequences. Energy efficiency as represented by total distance traveled (across all 100 note sequences) is examined. The results are shown in Table 7.

Table 7: Greedy vs Planned experiment 2 results.

	Normalized Distance
greedy	4.8
Viterbi	3.4

3.2.4 Discussion

The results demonstrate that there are benefits to planning versus on-line decision making. The planned movement sequences that are optimized for sequences of notes (such as a phrase) dropped much fewer notes and performed fewer octave jumps. The greedy method was slightly more energy efficient (according to distance traveled) in the first experiment. However, the second experiment was designed to explicitly evaluate distance traveled and energy efficiency in isolation. The system using Viterbi planning was roughly 30% more efficient.

One may argue that it is possible to develop a robot that is capable of playing everything and as a result doesn't require path planning or an inclusion of its physical identity as part of its note generating algorithms. This may be especially true for some percussion robots. For example, Eric Singer's *Orchestrion* system designed for Pat Metheny has a marimba playing robot that has a striker over each key allowing it to play any sequence of notes. Similarly, a disklavier has no need for a path planning algorithm that coordinates multiple motors.

Despite the advantages of these systems (and ignoring their disadvantages such as a paucity of social and visual cues), it can still be contended that incorporating a sense of physical identity is essential for new music to arise. While there are no limits on what may be played and such actuator per note systems can be sufficient for many musical applications, there are definitely opportunities that may be missed if any note generating system is implemented. For example, a note-to-note statistical model trained from human performances will inherently produce something playable

by people. Unless the robot knows it is capable of performing outside of what is capable by humans it will never take full advantage of its physicality. Techniques such as 15 note chord harmonies or simultaneous musical lines in six different octaves may have musical utility and beauty, but will not emerge from a musical intelligence lacking knowledge about its embodiment. Typically, composers and developers explicitly address these opportunities within the coding or composition process. In the next section I describe a generative music algorithm in which only the physical constraints are described and the intelligence leverages its understanding of the constraints to compose music fit for its physical self.

3.3 Embodied Generative Music

The integration of physical and musical components in a single state space for joint optimization is the paramount aspect of this work. Though Viterbi was the chosen algorithm it is likely that several search algorithms would have worked with some modifications to the state space. Ultimately, the efficacy of the search algorithm depends on two factors 1) the ability to represent a state such that it adequately encapsulates all relevant components and 2) the ability to evaluate states and state transitions with meaningful metrics.

The first factor was achieved with a representation that is relatively simple to interpret. The optimal path through the state lattice describes the sequence of physical configurations necessary to play the observed sequence of notes. The metrics were shown to be effective for the task of playing precomposed music, however, the musical heuristic (maximizing number of notes played) is not useful for generating music. In this section, the prospect of using a joint optimizing search methodology to generate music is explored.

3.3.1 Parametrized representations of higher-level musical semantics

Search and path planning requires that there is some information guiding the search towards good solutions (or what is deemed good according to a pre-defined metric). For the Viterbi algorithm this information includes the sequence of observations and the metric stating how well the path through the lattice explains or represents the observations. Previously, the observation sequence, Y, was a musical score, the observation space, O, constituted all the possible notes a score could contain, and the state space, S, constituted all the possible physical configurations of the robot.

For the system to create the note sequence as well as the movement sequence the state space must be expanded to include note information so that a single state combines a physical configuration and a specific note or notes to be played. The observation space and observation sequence need to be modified so that they encourage certain musical behaviors without explicitly dictating the notes to play. This is the 'musicianship' part of the problem.

The validity and efficacy of an integrated system relies on the assumption that there are multiple ways for a single higher-level musical idea or goal to be achieved. Without this assumption the methodology will fail. However, there is a lucid cogency to the assumption that presuming its truth is logically sound. For example, if a musician is asked to write a motif that ascends in pitch there are many possible motifs that can satisfy this constraint. The same holds true for more complex semantic musical features such as tension. Within jazz improvisation a musician may build tension by adding notes that fall outside of the scale harmony. This technique is used by many jazz musicians, yet, each musician is able to construct unique note sequences while still attaining the higher level goal of building tension. Furthermore, musicians are taught numerous other methods for building tension through manipulations of features such as dynamics, note density, or timbre. The premise is that a robot must rely on such musical semantics and have goals that govern individual note choices. A

single note sequence may be musically meaningful, but not necessarily possible when considering the robot's physical constraints, therefore, an alternative solution must be found. The purpose of this generative system is to find such a solution. Therefore, the observation space must encapsulate higher level semantics that can be successfully realized with more than one specific sequence of notes.

In this section a generative algorithm that explores embodied musical cognition within the context of jazz improvisation is discussed. The purpose of this section is not to evaluate the quality of the generated improvisations, but rather to demonstrate the utility of integrating the physical constraints into the decision process. **Figure 13** describes the general framework for the system (described in detail later on). It follows the idea that higher-level musical semantics are necessary.

This notion of dictating note sequences based on higher level musical features for jazz improvisation is markedly different from previous machine jazz improvisation systems. Typically, note-level models are developed using rule-based or statistical machine learning methods and any higher level musical intentions are an aftereffect [57]. Note-level models have been constructed using genetic algorithms, Markov chains, recurrent neural networks, and probabilistic grammars [18, 150, 64, 108]. Each of these systems uses note transitions as the building blocks for the generative algorithm. Though this may be sufficient for pure software applications, it is not a desirable approach for a robotic performer that may not be able to play what the system generates because it provides no higher-level context for finding more suitable alternatives.

Note-level models may be preferred because they are convenient for training statistical models. The system in this work does not utilize machine learning and instead employs knowledge-based heuristics that describe higher level musical concepts. Luckily, several musicians and academics have extensively studied the approaches of great jazz musicians over the last seventy or so years and have used their techniques

to define and teach modern jazz theory. This theory has been presented in numerous books and some of the higher level jazz concepts of this system are codified using these resources. Two books in particular (that have become canonical resources for jazz musicians) are the 'Jazz Piano Book' by Mark Levine and 'How to Improvise' by Hal Crook [128, 47]. The semantic concepts that are quantified in this system include harmonic tension as it pertains to scale theory, pitch contour, rhythmic complexity, and note density. The system chooses notes that prioritize the observation sequences describing these features while considering its physical constraints.

3.3.1.1 Harmonic Tension and Color

Arguably the most important feature for determining the pitches in a jazz solo is the chord progression. In fact, Johnson-Laird argues that the two most significant constraints for choosing pitches is the underlying chord progression and the contour considerations [96]. Mark Levine explains that in the early days of jazz, improvisers typically just embellished the melody of a tune and relied on the root, 3rd, 5th, and 7th of the underlying chord. It wasn't until the 1930s and 1940s when performers such as Bud Powell and Charlie Parker re-conceptualized the relationship between the notes they were playing and the chords these notes were being played over. Pitches were no longer thought of as the series of intervals that make up a chord, but rather degrees of a scale [128]. Gunther Schuller, historian and jazz musician, reinforces this claim with examples of improvisers deviating from the melody and constructing entirely new melodies based off the chord progression [172]. Harker also examines several solos of Louis Armstrong and provides examples of Armstrong's use of harmony-based rather than melody-based improvisation [76]. Finally, the significance of harmonic-based improvisation is further cemented by the state of jazz education today. In books (such as Mark Levine's) and in class curricula (such as Barrie Nettle's harmony course at Berklee College of Music) there is significant emphasis on harmony and scale theory.

At the most basic level, the essence of harmonic theory describes that a certain scale be played over a certain tonal center or more specifically a certain chord. For example, a 'C major' scale can be used to play over a *Cmaj7* chord or a 'G mixolydian' scale can be played over a *G7* chord. The 'C major' and 'G mixolydian' are actually comprised of the same notes as the *Cmaj7* chord is the 'I' chord (or tonic) of the C-tonal center and the *G7* functions as the 'V' chord (or dominant). In popular music the tonal center can usually be identified by just looking at the key signature of the piece. In jazz, however, the tonal center may modulate frequently throughout the course of a piece and the improviser must identify the tonal center given the chord sequence. It is not obvious what the tonal center is given a single chord because one chord can serve different harmonic functions depending on the tonal center. For example, a *D-7* can be the 'ii' chord of a progression with a tonic of 'C major' or it can function as the 'vi' chord of a progression with a tonic of 'F major'. Fortunately, there are common sequences of specific chord functions such as *ii-V-I, IV-V-I, or V-iv-i* that allow musicians to make reasonable assumptions where the tonal center lies. These sequences are referred to as cadences. Similarly to human interpretation strategies, a computational system can utilize this information to identify the tonal center.

To label the tonal centers and chord functions for each given chord a system can look for common cadences and find the optimal chord function sequence that maximizes the use of these cadences. This type of problem is referred to as a sequence tagging problem. It is similar to identifying the parts of speech of words in a sentence and, in fact, it is another type of problem well-suited for Viterbi decoding. There is no dataset to train a model to find the proper weights. The system described in the remainder of this section will instead tag sequences using hardcoded priors based on what is most frequently used in classical and jazz music.

In this case, the state space is comprised of tuples consisting of all possible

combinations of tonal centers and chord functions. Scales representing each mode derived from Major, Melodic Minor, Harmonic Minor, Harmonic Major, and Double Harmonic Major harmonies are defined. This means the system knows 35 scales, however, in tonal music stemming from classical and jazz only a small subset of these scales can serve as tonics. This is because in typical harmonies of tonal music the dominant triad is a major or minor chord. This limits the number of scales that can be used as the basis for the harmony. All the scales the system knows are listed below and the possible tonics are highlighted in bold.

1. Major Scale Harmony
    (a) **Ionian - Possible Tonic**
    (b) Dorian
    (c) Phrygian
    (d) Lydian
    (e) Mixolydian
    (f) **Aeolian - Possible Tonic**
    (g) Locrian
2. Melodic Minor Harmony
    (a) **Minor Major - Possible Tonic**
    (b) Dorian b9
    (c) Lydian Augmented
    (d) Lydian Dominant
    (e) **Mixolydian b6 - Possible Tonic**
    (f) Half Diminished (Locrian #2)
    (g) Altered (Super Locrian)
3. Harmonic Minor Harmony
    (a) **Harmonic Minor - Possible Tonic**

(b) Locrian #6

(c) Ionian Augmented

(d) Romanian

(e) Phyrgian Dominant

(f) Lydian #2

(g) Ultra Locrian

4. Harmonic Major Harmony

(a) **Harmonic Major - Possible Tonic**

(b) Dorian b5

(c) Phrygian b4

(d) Lydian b3

(e) Mixyolydian b9

(f) Lydian Augmented #2

(g) Locrian bb7

5. Double Harmonic Major Harmony

(a) **Double Harmonic Major - Possible Tonic**

(b) Lydian #2#6

(c) Ultra-Phrygian

(d) **Hungarian Minor - Possible Tonic**

(e) Oriental

(f) Ionian Augmented #2

(g) Locrian bb3 bb7

Each of the eight tonics has seven possible scale degrees meaning there are seven chord functions within that harmony that can be played. These can be played in 12 keys making the total number of possible states equal to 672 ($8 \times 7 \times 12$). The transition matrix is 3-dimensional (to include cadences with a length of three) and

gives weight to commonly seen cadences. These weights are not learned, but manually labeled using expert knowledge and then evaluated with a test set of ten jazz standards.

Table 8: Frequently used chord sequences with manually labeled transition weights.

Scale	Sequence	Weight
Ionian	IV-V-I	3
	ii-V-I	3
	V-ii-I	3
	iii-ii-I	1
	iii-IV-I	2
	V-I	3
	IV-I	2
	ii-V	3
	IV-V	3
	I-vii°	2
Aeolian	v-iv-i	3
	bVII-iv-i	3
	bVI-bVII-i	3
	iv-bVII-i	3
	bVI-v-i	2
	iv-v-i	1
	bVII-i	3
	v-i	1
Harmonic Minor	ii°-V-i	3
	iv-V-I	3
	V-iv-i	2

Table 8 Continued.

	bVI-V-i	2
	v-bVI-i	2
	V-i	3
	ii°-V	3
	i-vii°	2
	i-V	2
Harmonic Major	iv-V-I	3
	V-iv-I	3
	iii-iv-I	2
	V-I	2
Minor major	IV-V-i	3
	ii-V-I	3
	V-I	2
	I-vii°	2
Mixolydian b6	v-iv-I	2
	bVII-iv-I	2
	bVIII-I	2
Double Harmonic	iii-bII-I	1
Major	bII-I	1
Hungarian Minor	bVI-vii-i	1

The observation space is made up of the chord progression and melody of a piece. If there is no melody and only the chord progression is provided there is no instantaneous score to evaluate a single chord. However, when a melody is present the emission

matrix describes the relationship between the notes in the melody that are played over a given chord. The score is derived by counting the number of melody notes that are part of the scale for the tonic and chord function being evaluated.

Finally, using the chord function state space, a transition matrix describing common cadences, and an emission matrix describing note/scale relationships Viterbi can be used to find the optimal sequence through the chord function and tonal center state space. With a small test of ten jazz standards with labeled chord functions (labeled by one expert) it is possible to achieve greater than 95% agreement between the Viterbi tagged chord functions and those hand labeled in the test set. However, there is some subjectivity when it comes to tagging these sequences. Two experts are likely to agree on the majority of functions a specific chord serves in the context of a piece, but will also likely differ in some places. Differing interpretations can be useful and emphasis on certain centers and harmony is likely to be a key a component describing an individual's style. An example of the sequence tagging is shown in **Figure 10**.

Understanding the tonal context of specific chord sequences in an entire progression is the first part of modeling tonal tension. The second part evaluates different pitches used with the specific chords and tonal centers. When human improvisers and composers evaluate individual pitches against a certain chord they are considering its context in the scale that is being used with the chord [128, 47]. The tonal center and chord function determines the scale and with this information the improviser can evaluate the tonal quality of individual pitches relative to the scale and chord. For example, in a *D-7, G7, Cmaj7* (a major *ii-V-I*) progression the D-dorian, G-mixolydian, and C-ionian scales are the scales that represent the C-major harmony for each chord in the sequence, respectively. Certain pitches are more likely to create dissonance when played over this progression. It is up to the musician to be aware of this and resolve the tensions to pitches that support the underlying tonal centers. For the *Cmaj7* chord, stable notes of the ionian scale include the root, 3rd, 5th, and 7th. This

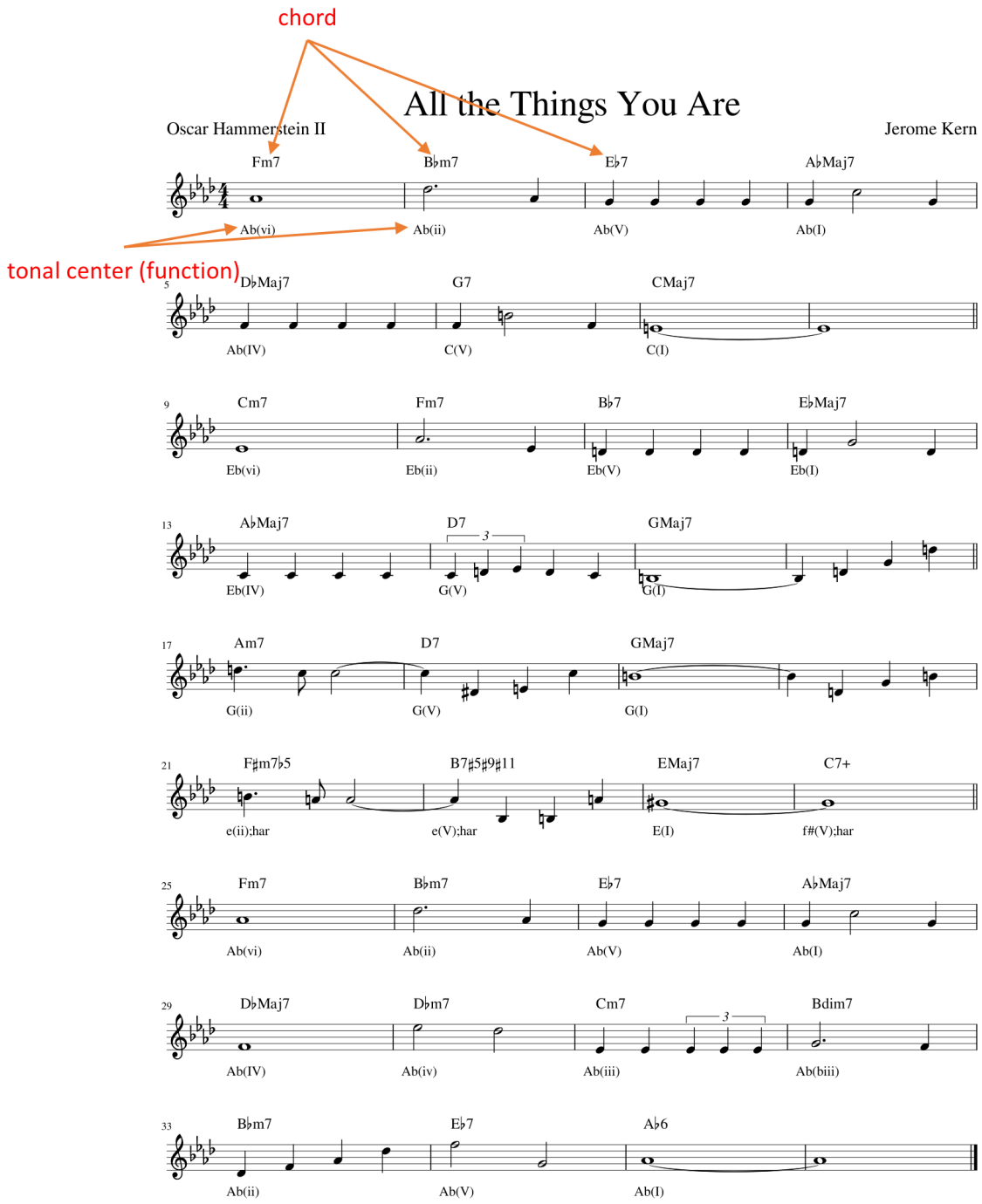


Figure 10: The chords (annotated above the measure) are tagged with their corresponding tonal center (annotated below the measure) using automatic sequence tagging based on a trigram model. Viterbi decoding is used to find the optimal sequence.

is not surprising because these are the notes that make up the chord so, by definition, they are harmonically stable. The 2nd and 6th scale degrees (or 9th and 13th in terms

of chord degrees) are known as available tensions that are still harmonically stable, but can add some interesting color to the melodic line. The 4th scale degree (the 'F' in C-ionian) and pitches outside of the C-ionian scale are considered especially dissonant relative to the chord. If the objective of the improviser is to emphasize the underlying tonal center (in jazz vernacular this is referred to as "playing the changes") then these notes should be resolved to harmonically stable pitches via techniques such as voice leading or chromatic steps.

This model is used to measure the tonal distance between a pitch and a chord. The model can be thought of as a nonlinear projection of the pitch space (similar to the circle of 5ths or tonnetz models) based on heuristics provided by scale theory and the current tonal center. See **Figure 11** for an outline of the model. The tonal model produces five levels of harmonic distance a pitch can have from a given chord from closest to furtherest this includes: (1) the root pitch; (2) pitches other than the root that are in the chord; (3) available pitch tensions that can be used with the chord; (4) avoid pitches in the scale represented by the tonal center and chord; (5) all pitches outside of the scale represented by the tonal center and chord. These five levels are used to measure 'tonal color' or harmonic tension of an individual pitch at any given moment within the chord progression. Given a time series or observation sequence of tonal color values the system will find the best path that explains the sequence.

3.3.1.2 Pitch Contour

The second constraint for determining the pitch is the overall contour and octave location. Here, the observation sequence is a time series representing the absolute pitch. The path planning will create a path that best describes this sequence. The emission metric used is simply the distance between the state's note and the value of the observation in absolute pitch space. The transition metric is simply a descending, ascending, or non-moving descriptor between two notes in the state space. There is a

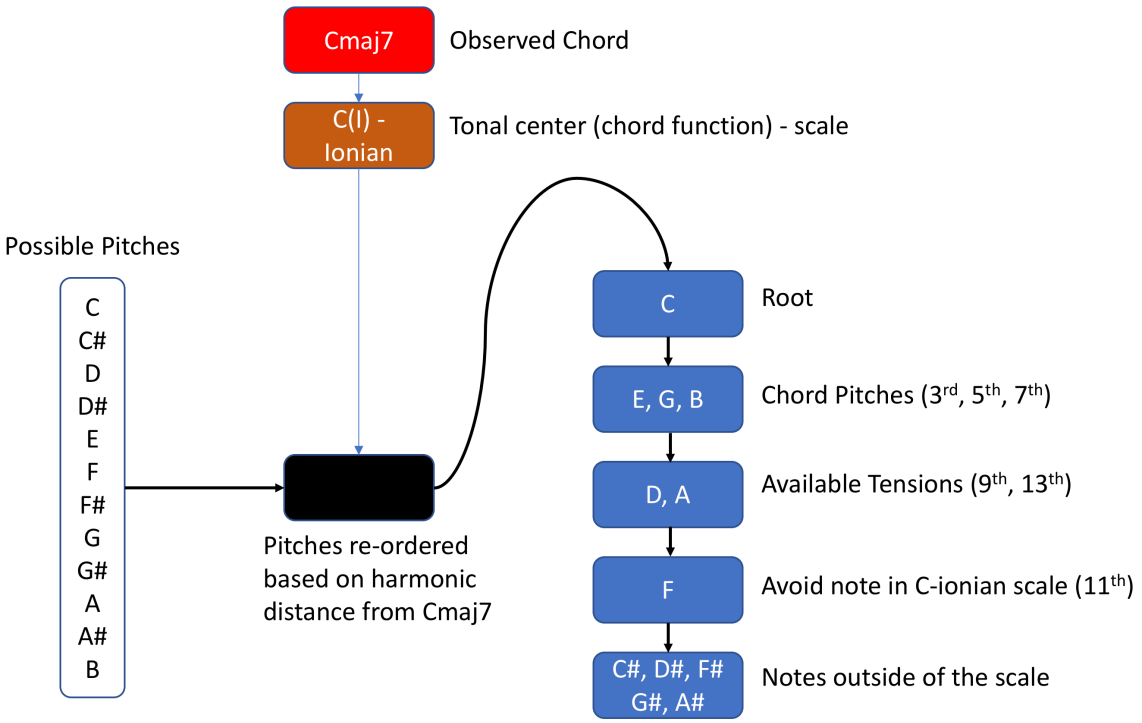


Figure 11: Harmonic distance metric - The observed chord (Cmaj7) is labeled by the chord function Viterbi decoding process. The possible pitch classes are measured in terms of harmonic distance (or harmonic stability) relative to the particular chord function and tonal center. Using harmonic theory, the pitches are projected into a space that organizes them according to their stability over the chord function determined by the Viterbi process. There are five levels in this space and the level number is used to represent the harmonic distance of the pitch to the given chord. In this example, the root note, *C*, is the closest note to the 'Cmaj7' chord and the notes outside of the C-ionian scale are considered the furthest.

small penalty if the transition between two states does match the transition between the corresponding two observation states.

3.3.1.3 Rhythm

The note choices are also constrained by aspects of rhythm. The two factors used in this work to describe rhythm are note density and complexity. Rhythm choices are made on a per beat basis. The system contains hundreds of units encapsulating different rhythms that can be played within the duration of a single beat (**Figure 12**). These units were taken from examples in George Lawrence Stone's classic instructional

drumming book *Stick Control: For the Snare Drummer* [191]. Note density describes the concentration of notes played within a finite period of time. The observation space for the note density feature represents this concentration per beat.

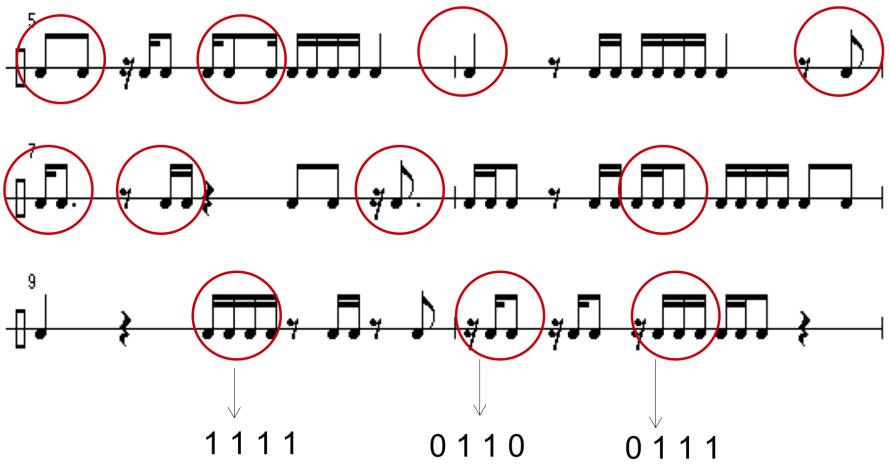


Figure 12: Rhythmic decisions are made per beat. A library of hundreds of unique rhythmic units exist and the system pulls from this library. In this figure the unique rhythms are circled. The rhythms can be encoded as binary strings encoding only the onset and ignoring duration. They are shown as having a resolution of one sixteenth note, but the implementation provides up to 128th note resolution containing both duples and triples. The decision to remove durations was to reduce the dimensionality of the task and focus on percussionist robots such as Shimon.

The complexity of an object describes the amount of information encoded in it. Therefore, complexity of a system can be measured by how much it can be compressed. One common way of measuring rhythmic complexity is to use Lempel-Ziv compression on a binary string [180, 222, 180, 178, 179, 196]. Similarly, a string can be represented as a finite state machine (FSM). The lower bound of the output length of a sequence generated by a finite state machine is equal to the upper bound of the output length of Lempel-Ziv algorithm. Here, a rhythmic unit is encoded as an FSM and the size of the FSM represents the complexity of the rhythm. This method can be thought of as a baseline for rhythmic complexity. It is capturing hierarchical redundancies encapsulated in the rhythmic times series, but does not specifically address the subjectivity and human perceptions of rhythmic complexity. For example,

people often correlate rhythmic complexity with how difficult it is perceived to play rather than the hierarchical structure. However, it has been shown that this method of encoding does correlate, to a degree, with human perception and that is why it is considered a baseline [196].

In this implementation a single rhythmic unit complexity is measured using the value described by the size of the state machine. The transition between units can also be measured by creating an FSM of the entire concatenated rhythmic string across both rhythmic states in the transition. Because rhythms are represented in predefined units their complexity and note density can be computed a priori. Additionally, rhythmic features are tempo dependent when considering the physical constraints of the robot. The rhythmic unit library can be pruned prior to path planning by removing impossible rhythms according to the physicality.

3.3.2 Joint Optimization

In Section 3.2 the Viterbi algorithm was used to find the physical movement sequence necessary to play a sequence of notes, Y, that was provided prior to the search. Here, there exists an observation sequence for each musical semantic (as shown in **Figure 13**) and Viterbi is used to find the optimal note sequence derived from both the musical parameters and physical constraints. Therefore, the emission and transition matrices not only constrain decisions to the robot's embodiment, but also evaluate the note-to-note decisions based on their ability to satisfy the semantic goals.

Pitch contour, harmonic color, note density, and rhythmic complexity are the core semantics driving the decisions during the path planning process. These were chosen based off of Johnson-Laird's description of how musicians improvise [96]. He argues that successful melodies can be generated by prioritizing individual pitches based on the constraints of the chord functions and contour with appropriate rhythmic figures.

Each of these semantics is represented as a time series. Each set of time series

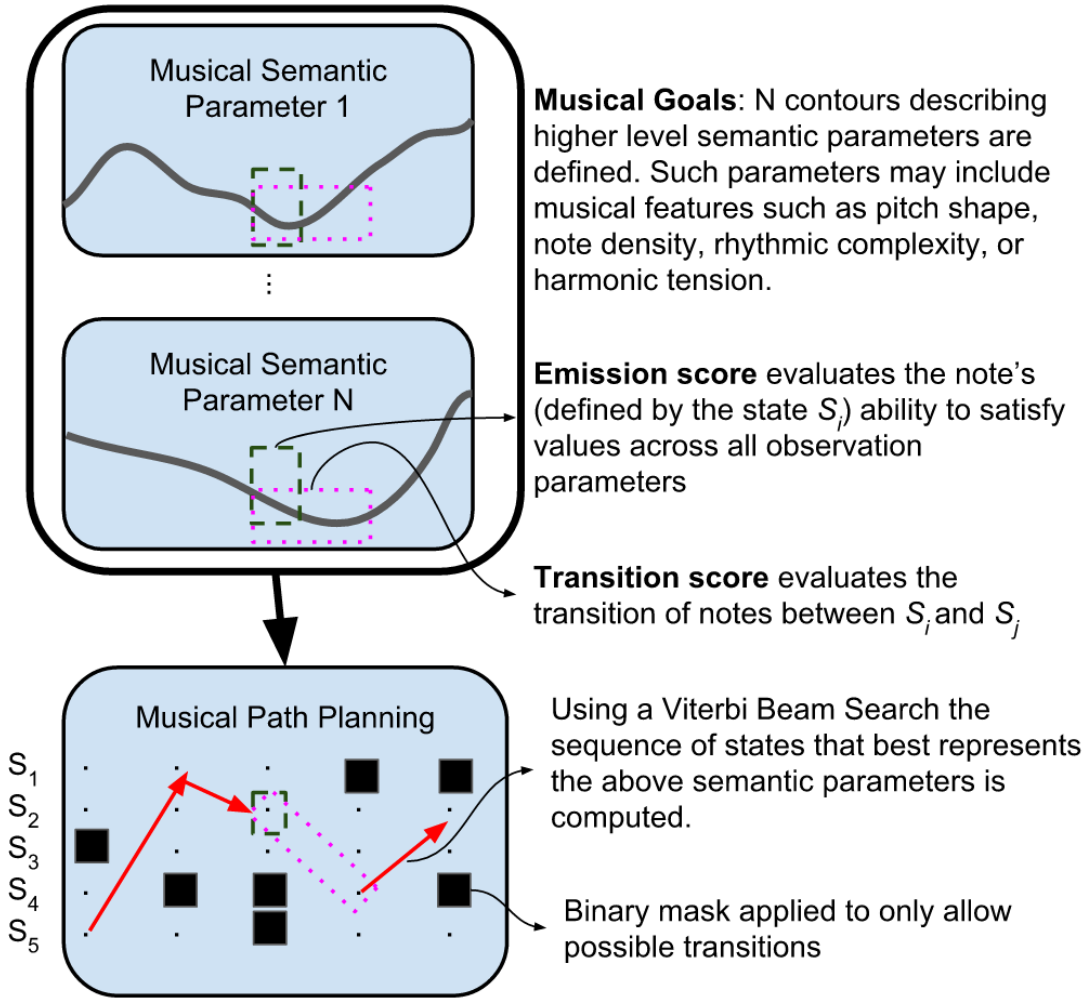


Figure 13: Framework of the generative embodied music system.

represents the duration of an individual phrase that is defined by the user in numbers of beats. The path planning generates the sequence that jointly describes all the semantics. A parameter weighing one semantic over another can be applied. A GUI (using MaxMSP) gives a user the ability to create observation sequences of each semantic and its weight (**Figure 14**).

Each state in S is now defined by a configuration and a note to be played. For Shimon, the state space includes about 73,815 possible arm configurations. This state space is likely to vary considerably depending on the robotic platform. For platforms with enormous state spaces finding the global optimal path is not always feasible if

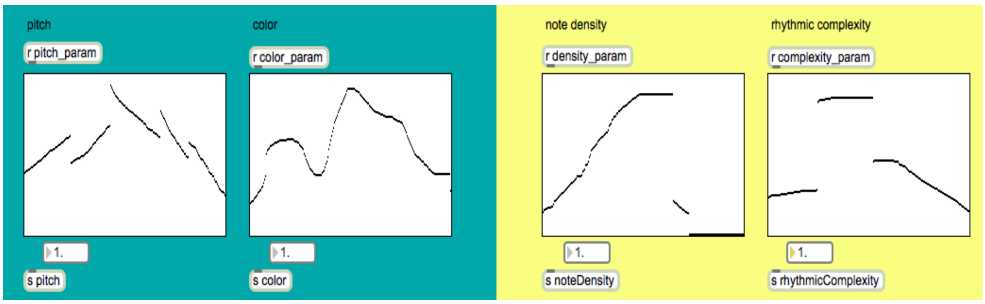


Figure 14: GUI for drawing contours of pitch contour, harmonic color, note density, and rhythmic complexity.

the path is to be found in a timely manner. Therefore, a beam search is applied in order to prune unreasonable branches quickly. Additionally, the Viterbi algorithm lends itself to distributed computing techniques and can be computed across multiple CPUs if the state spaces are very large.

For each musical semantic, p_n (note density, rhythmic complexity, etc.) a time series of observations is provided (assume these time series are manually provided to the system). The emission score for state s_i at observation time t describes the aggregate score across all N parameters

$$b_t = \sum_{n=0}^{N} \lambda_n R(p_{n_t}, s_i) \tag{7}$$

where $R(p_n, s_i)$ describes the instantaneous score of the note in s_i given the semantic parameter and λ_n is applied to each parameter to describe its weight within the overall score. The transition score is described as

$$a_t = \sum_{n=0}^{N} \lambda_n R(p_{n_t}, p_{n_{t-1}}, s_t, s_{t-1}) \tag{8}$$

where $R(p_{n_t}, p_{n_{t-1}}, s_t, s_{t-1})$ is the score describing how well the transition between notes in the states s_t and s_{t-1} represent the transition between p_{n_t} and $p_{n_{t-1}}$. The optimal path is computed given these emission and transition scores for the semantic parameters.

3.3.2.1 Generation

In order to automatically generate the music (without a person drawing the semantic contours) a multivariate Markov chain is used. Instead of creating a Markov chain at the note level (as is typical), these chains are learned at the semantic level. The goal is for the computer to autonomously generate the semantic contours. Over time many contours were manually drawn using the GUI to generate different phrases. These contours were saved and served as the dataset to learn the Markov probabilities. In total there are 88 sets of contours in the dataset, where a single set of contours includes the four semantic observation sequences.

The multivariate Markov chain is used to capture both intra and inter-transition probabilities across multiple sequences. In this case, the multivariate Markov chain generates the sequences for all features simultaneously. The hypothesis is that the features are correlated in some way and without capturing these relationships in a multivariate method then these correlations would be lost.

The implementation is based off of the work by Ching et al. [36] and formalized as:

$$\mathbf{X_{r+1}} \equiv \begin{pmatrix} \mathbf{x}_{r+1}^{(1)} \\ \mathbf{x}_{r+1}^{(2)} \\ \vdots \\ \mathbf{x}_{r+1}^{(s)} \end{pmatrix} = \begin{pmatrix} \lambda_{11}P^{(11)} & \lambda_{12}P^{(12)} & ... & \lambda_{1s}P^{(1s)} \\ \lambda_{21}P^{(21)} & \lambda_{22}P^{(22)} & ... & \lambda_{2s}P^{(2s)} \\ \vdots & \vdots & \vdots & \vdots \\ \lambda_{s1}P^{(s1)} & \lambda_{s2}P^{(s2)} & ... & \lambda_{ss}P^{(ss)} \end{pmatrix} \begin{pmatrix} \mathbf{x}_{r}^{(1)} \\ \mathbf{x}_{r}^{(2)} \\ \vdots \\ \mathbf{x}_{r}^{(s)} \end{pmatrix} \tag{9}$$

The state probability distribution of the j_{th} sequence, $x_{r+1}^{(j)}$ at the time $(r+1)$, depends on the weighted average of $P^{(jk)}\mathbf{x}_r^{(k)}$ where $P^{(jk)}$ is the transition probability matrix from the states at time t in the k_{th} sequence to the states in the j_{th} sequence at time $t+1$, and $\mathbf{x}_r^{(k)}$ is the state probability distribution of the k_{th} sequence at the time r. By collecting a database of these musical observation sequences it is possible perform this matrix multiplication and generate the contours.

3.3.3 Musical Results

The system is used to generate music under different conditions. The objective is to demonstrate that alternative musical decisions are made as a result of the physical constraints. **Figure 15** shows musical outputs in which the semantic observation sequences are static, but the physical constraints are modified (see video for audio examples[1]). Thus, the differences among the outputs are a result of physical design and unique embodiment.

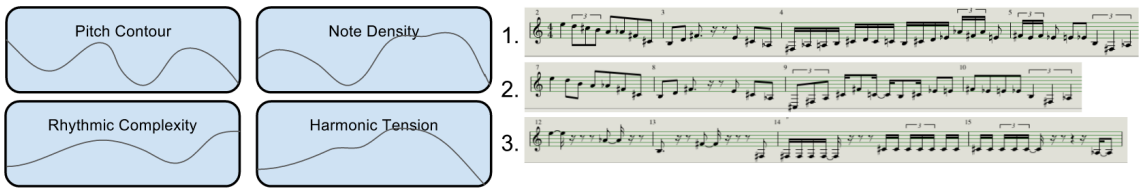


Figure 15: Embodied generation samples. Three motifs are generated to satisfy the musical parameters on the left using different physical constraints. **1.** A single arm robot that can move at a fast rate of one half step per millisecond. **2.** Robot with four larger and slower arms (Shimon-like setting) that must avoid collision. **3.** Very slow moving single arm robot with a fast strike rate of 20hz.

When applied to a real robotic platform the system performs adequately by generating note sequence the robot is capable of performing. In **Figure 16** an excerpt from a generated solo using the physical constraints of the Shimon robot is shown (the entire solo can be found in Appendix A). Notice that during the faster sixteenth riffs the pitch intervals are generally larger than those of the eighth note sequence riffs. This is because it is physically impossible for Shimon to play such small intervals at the faster rate. Instead, the system generates the best sequence that approximates the musical features given Shimon's physical constraints. The higher note density motifs demonstrate arpeggio-like sequences and the pitch contours are a bit more exaggerated and extreme. Recall that Django Reinhardt made similar adjustments and adaptations addressing the loss of his two fingers.

[1] `https://www.youtube.com/watch?v=S2Yda1ndKVc`

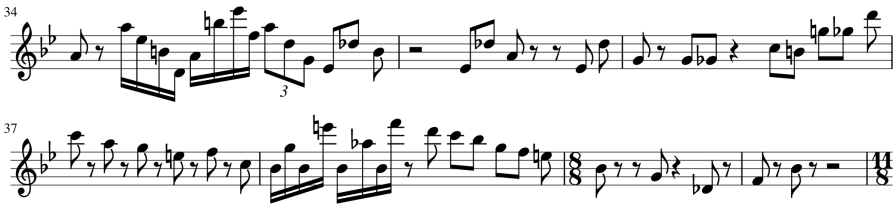


Figure 16: An excerpt from a solo generated for the physical constraints the Shimon robot.

In **Figure 17** an excerpt from a generate solo using the physical constraints designed to roughly emulate a human vibraphone player is shown. The physical constraints allow for two arms and the allowable be interval between successive notes of one arm is set to 150ms, thus, using two arms the system can play rhythms with onset intervals greater than or equal to 75ms.

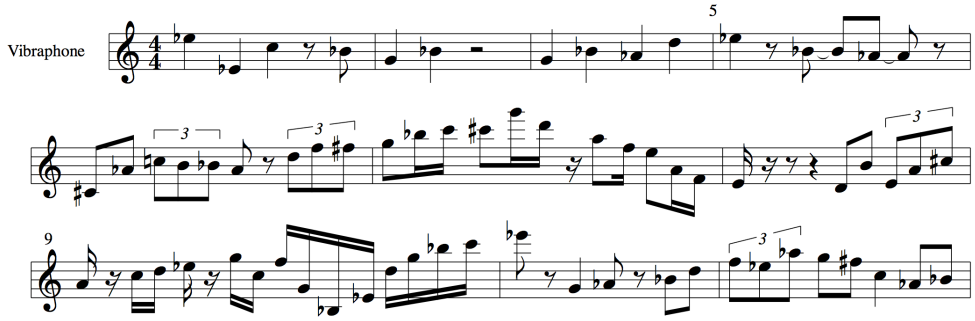


Figure 17: An excerpt from a solo generated over the chord progression from the jazz standard "All the Things You Are" for a set of simulated physical constraints emulating a human vibraphone player.

While there are signs of stylistic biases and emergence resulting from the parameters used for the examples above, to demonstrate the effect of physical constraints on the musical output we can simulate an extremely capable robot. In **Figure 18**[2] an example of a solo generated by a set of simulated physical constraints for a hypothetical robot is shown. The physical constraints allow for incredibly fast movement and access to a wide pitch range. The solo was generated for a tempo of 115 bpm and the resulting

[2]This figure shows notes played simultaneously, but this is an artefact resulting from the score editor's inability to represent the necessary temporal resolution. The generated music is monophonic.

music produces motifs and riffs that are humanly impossible. This musical output would never have been generated if the system did not have some understanding of its physicality (unless the developer or composer explicitly encouraged these types of behaviors).

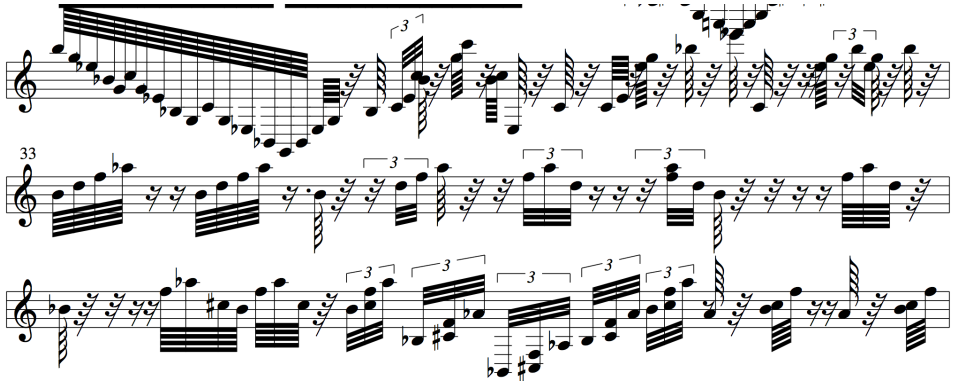


Figure 18: An excerpt from a solo generated over the chord progression from the jazz standard "All the Things You Are" for a set of simulated physical constraints given a hypothetical robot a wide range of capabilities.

3.3.4 Discussion

The jazz composition system presents a proof of concept that physical constraints can influence the musical decisions. While this is qualitatively true, the nature of this particular system poses some challenges.

Perhaps the most significant caveat of this system is that it requires planning. Having developed many real-time interactive music systems, I understand the appeal from a developer's perspective of greedy decision-making processes and systems that allow for notes to be generated in an on-line fashion. To take advantage of Viterbi and a planning process, multiple notes need to be processed at once. However, the use of planning makes sense on multiple levels. If all important musical features could be described by extremely localized characteristics without any regard to longer term structure a simple n-gram model would suffice as the solution to music generation and perfect style modeling. This is not the case, however, and is demonstrated as such from

how people talk and think about music. Norgaard explains the importance of planning as a key component of the thinking processes of artist-level human improvisers [151]. Not only does planning exist, but it is linked to musical expertise. Fidlon shows that more expert musicians were planning their musical strategies much further into the future compared to less capable musicians [62]. Recently, Norgaard demonstrated similar findings. Developing musicians tended to make note-level decisions, while the artist-level musicians planned out ideas that stretched over entire sections [153].

Additional elements of human improvisational thinking support the notion of planning over multiple notes as well. Often jazz musicians describe their decisions as stemming from an initial pool of musical ideas [151]. This "idea bank" consists of higher level musical semantics encapsulating sequences of notes. This is evidenced through the repeated use of specific licks and riffs by artist-level musicians and their modifications for different harmonic contexts.

These examples of human thinking suggest planning can be useful for generating optimal sequences from a purely musical perspective, but planning concepts have additional relevance to robotic musicianship applications compared to pure software applications. In software, sounds can be generated instantaneously, however, in the natural world some energy must be provided to the system to create sound acoustically. In robotic musicianship, the robot supplies the necessary kinetic energy to create a sound, thus, requiring some form of movement. If the robot really does have noticeable constraints, there will be inherent delays resulting from this movement. Most good solenoid-based striking or fast motor-based striking mechanisms do not have a noticeable delay between the time the computer sends the message to strike and the system strikes and produces sound. However, if additional motions are necessary, as is the case with Shimon, a noticeable delay may become present. Though it is possible for the delay to be very small, in practice these types of motions usually involve motors moving heavier components (like an entire limb) that can introduce

delays of hundreds of milliseconds. Therefore, for real-time systems, some planning is already required in order for the robot to adhere to specific beat markers.

Another challenging characteristic of this system is the enormous state space. While Viterbi lends itself to a relatively straightforward distributed computing implementation, without significant hardware resources it is impossible to compute the paths in real-time. Alternative (and perhaps better) solutions to speeding up computation require heuristically driven pruning of the state space. It is likely that the states can be prioritized according to both musical and physical heuristics. This idea is addressed in the next chapter.

3.4 Conclusion

In this chapter a musical path planning method based on Viterbi beam search was described. The two considerations for the algorithmic design were optimality and musical emergence. In the first part of the chapter it was shown that a planning method can create more optimal movement plans compared to a greedy method allowing the robot to play a precomposed sequence of notes. This was evaluated using metrics describing the musical deviation from the original sequence (how many notes dropped or transposed) and energy efficiency (represented by total distance traveled).

In the second part of the chapter it was shown that by jointly optimizing between a system's physical constraints and musical reasoning and decision processes different musical behaviors emerged. Using a set of heuristics derived from jazz theory the system generated note sequences to support the heuristics while accommodating for various types of physical constraints.

Finally, this system has been extensively demonstrated and used in multiple performance settings. The path planning mechanism serves as the underlying technology for Shimon to perform precomposed melodies. The jazz system serves as the generative technology that Shimon uses to create its own music and has been used to generate

all of its solos for the *Shimon and Friends* concert series. This includes performances at The Kennedy Center in Washington, D.C., at Snug Harbor Cultural Center in New York, and at the Shanghai International Interactive Arts Festival (for descriptions about the concert series and a full list of performances see Appendix B).

CHAPTER IV

LEARNING MUSICAL SEMANTICS

Sections from this chapter have been prepared and accepted for publication in:

> **Bretan, Mason, Gil Weinberg, and Larry Heck. "A Unit Selection Methodology for Music Generation using Deep Neural Networks." ICCC, 2017.**

In the previous Chapter the musical heuristics were derived from concepts described by experts and then codified. This chapter explores whether meaningful musical semantics can be learned from data and then applied to autonomous generation. By learning the heuristics it is potentially possible to capture relevant musical information or featuresa that an expert may overlook or not know how to codify. The proposed method leverages unit selection and concatenation as a means of generating music using a procedure based on ranking, where, a unit is considered to be a variable length number of measures of music. First, we examine whether a unit selection method, that is restricted to a finite size unit library, can be sufficient for encompassing a wide spectrum of music. This is done by developing a deep autoencoder that encodes a musical input and reconstructs the input by selecting from the library. Then a generative model combining a deep structured semantic model (DSSM) with a long short term memory (LSTM) network to predict the next unit is described. This model is evaluated using objective metrics including mean rank and accuracy and with a subjective listening test in which expert musicians are asked to complete a forced-choiced ranking task. The model is compared to a note-level generative baseline that consists of a stacked LSTM trained to predict forward by one note. Finally, a method for incorporating the physical parameters of an embodied system is described.

The embodied process uses a re-ranking strategy based on a metric defined by the Viterbi path planning process.

4.1 Introduction

In Chapter 3, a generative system based on knowledge-based heuristics was described. While knowledge-based systems can produce compelling results, they are constrictive in that they can only represent what is codified by the developer. Though an expert musician should be able to build an expert system, there are important features in music that are likely to go unaddressed. Music is an extremely high dimensional domain and capturing all relevant features using a combination of rules and various hand-designed heuristics is not possible. Therefore, an alternative methodology to creating the heuristics is desirable.

For the last half century researchers and artists have developed many types of algorithmic composition systems. Many of these efforts are driven by the allure of both simulating human aesthetic creativity through computation and tapping into the artistic potential deep-seated in the inhuman characteristics of computers. Some systems may employ rule-based, sampling, or morphing methodologies to create music [159]. In this chapter, I present a method that falls into the class of symbolic generative music systems consisting of data driven models which utilize statistical machine learning.

Within this class of music systems, the most prevalent method is to create a model that learns likely transitions between notes using sequential modeling techniques such as Markov chains or recurrent neural networks [156, 65]. The learning minimizes note-level perplexity and during generation the models may stochastically or deterministically select the next best note given the preceding note(s). However, there is significant evidence that musical decisions are made using collections of predetermined note groupings [163], in other words, instead of making a decision about one note

at a time, humans make decisions about groups of notes are made and inserted into the performance. In a recent study, Norgaard analyzed a collection of Charlie Parker solos and concluded, "the sheer ubiquity of patterns and the pairing of pitch and rhythm patterns support the theory that preformed structures are inserted during improvisation. The patterns may be encoded both during deliberate practice and through incidental learning processes" [152]. However, from his qualitative investigation based on artist-level interviews he found that improvisation is likely a combination of inserting well-learned ideas from memory and tweaking those ideas to fit the specific harmonic context. Here, a system that uses a similar combination approach is presented and a method to generate monophonic melodic lines based on unit selection is outlined. In this work a unit is a precomposed section of music with a fixed duration such as one or two measures.

This first part of this chapter focuses solely on effectively learning a musical space without addressing any physical constraints. The second part of chapter describes a method for applying what is learned using a disembodied generative music approach. The last part of the chapter addresses how this method can be augmented to address the physical constraints of a robotic system such that units are chosen according to music as well as physical related metrics. The three immediate research objectives include:

1. **Establish a metric to describe musical units** — Unlike work described in the previous chapter in which knowledge-based musical heuristics were used to generate a musical sequence, here, the objective is to learn the features using deep neural networks. A successful representation of a unit or one of the preformed structures to which Norgaard refers, is one that can be concisely described within a vector space that captures perceptually meaningful traits.

2. **Develop a method for selecting and concatenating units** — For generating sequences it is necessary to be able to do more than just numerically describe

a unit. Adjacent units must be semantically similar and seamlessly connect together based on a combination of musical priors and the physical constraints of the performer.

3. **Modify pitches in units by jointly optimizing for harmonic context and physical constraints** — For a robotic musician to use unit selection it must be able to play the units. Therefore, path planning is used as a final step in the process. The units themselves serve as a sort of heuristic that can help to bias the types of decisions the path planning will make, thus, making it possible to prune the state space and make decisions in a timely manner.

The architecture of this work is inspired by a technique that is commonly used in text-to-speech (TTS) systems. The two system design trends found in TTS are statistical parametric and unit selection [223]. In the former, speech is completely reconstructed given a set of parameters. The premise for the latter is that new, intelligible, and natural sounding speech can be synthesized by concatenating smaller audio units that were derived from a preexisting speech signal [88, 19, 43]. Unlike a parametric system, which reconstructs the signal from the bottom up, the information within a unit is preserved and is directly applied for signal construction. When this approach is applied to music, the generative system can similarly get some of the structure inherent to music "for free" by pulling from a unit library.

The ability to directly use the music that was previously composed or performed by a human can be a significant advantage when trying to imitate a style or pass a musical Turing test [46]. However, there are also drawbacks to unit selection that the more common note-to-note level generation methods do not need to address. The most obvious drawback is that the output of a unit selection method is restricted to what is available in the unit library. Note-level generation provides maximum flexibility in what can be produced. Ideally, the units in a unit selection method should be small enough such that it is possible to produce a wide spectrum of music, while remaining

large enough to take advantage of the built-in information.

Another challenge with unit selection is that the concatenation process may lead to "jumps" or "shifts" in the musical content or style that may sound unnatural and jarring to a listener. Even if the selection process accounts for this, the size of the library must be sufficiently large in order to address many scenarios. Thus, the process of selecting units can equate to a massive number of comparisons among units when the library is very big. Even after pruning the computational demands can be high. However, the method can be effective as long as the computing power is available and unit evaluation can be performed in parallel processes. Additionally, methods such as vector quantization can be applied to reduce the number of comparisons in the nearest neighbor search.

In the first part of this chapter unit selection as a means of music generation is explored. First, a deep autoencoder is developed in which reconstruction is performed using unit selection. This allows us to make an initial qualitative assessment of the ability of a finite-sized library to reconstruct never before seen music. Then, a generative method that selects and concatenates units to create new music is described.

The proposed generation system ranks individual units based on two values: 1) a semantic relevance score between two units and 2) a concatenation cost that describes the distortion at the seams where units connect. The semantic relevance score is determined by using a deep structured semantic model (DSSM) to compute the distance between two units in a compressed embedding space [86]. The concatenation cost is derived by first learning the likelihood of a sequence of musical events (such as individual notes) with an LSTM and then using this LSTM to evaluate the likelihood of two consecutive units. The model's ability to select the next best unit is evaluated based on ranking accuracy and mean rank. To measure the subjective nature of music and evaluate whether the networks have learned to project music into a meaningful space a listening study is performed. The study evaluates the "naturalness" and

"likeability" of the musical output produced by versions of the system using units of lengths four, two, and one measures. Additionally, these unit selection based systems are compared to the more common note-level generative models. As a baseline an LSTM trained to predict forward by one note is used.

4.2 Related Work

Many machine learning based methods for generating music have been proposed. The data-driven statistical methods typically employ n-gram or Markov models [37, 156, 204, 181, 42]. In these Markov-based approaches note-to-note transitions are modeled (typically bi-gram or tri-gram note models). However, by focusing only on such local temporal dependencies these models fail to take into account the higher level structure and semantics important to music.

Like the Markov approaches, RNN methods that are trained on note-to-note transitions fail to capture higher level semantics and long term dependencies [40, 20, 70]. However, using an LSTM, Eck demonstrated that some higher level temporal structure can be learned [58]. The overall harmonic form of the blues was learned by training the network with various improvisations over the standard blues progression.

However, a note-to-note level model is likely not adequate for communicating larger scale complex musical ideas. A note-level model trained to predict the next note can describe the likelihood of a note sequence, but beyond this contains little musical relevance. A melody (precomposed or improvised) relies on a hierarchical structure and the higher-levels in this hierarchy are arguably the most important part of generating a melody. Much like in story telling it is the broad narrative arcs that are of the most interest and not necessarily the individual words. Ideally, a model should concisely encapsulate these higher level musical semantics (such as pitch contour, harmonic color, rhythmic complexity, and note density). Furthermore, the previous chapter demonstrated the utility of planning in robotic musicianship. Choosing notes

one at a time limits the ability of a robotic system from making a decision that may have adverse effects in the future.

Rule-based grammar methods have been developed to address such hierarchical structure. Though many of these systems' rules are derived using a well-thought out and careful consideration to music theory and perception [124], some of them do employ machine learning methods to create the rules. This includes stochastic grammars and constraint based reasoning methods [140]. However, grammar based systems are used predominantly from an analysis perspective and do not typically generalize beyond specific scenarios [126, 159].

The most closely related work to our proposed unit selection method is David Cope's *Experiments in Musical Intelligence*, in which "recombinancy" is used [45]. Cope's process of recombinancy first breaks down a musical piece into small segments, labels these segments based on various characteristics, and reorders or "recombines" them based on a set of musical rules to create a new piece. Though there is no machine learning involved, the underlying process of stitching together preexisting segments is similar to the proposed method. However, the work described in this chapter attempts to create effective labels for each unit using a a semantic embedding derived a technique developed for ranking tasks in natural language processing (NLP). Furthermore, the concatenation process is based on sequential modeling using an LSTM.

The goal in this research is to examine the potential for unit selection as a means of music generation. Ideally, the method should capture some of the structural hierarchy inherent to music like the grammar based strategies, but be flexible enough so that they generalize as well as the generative note-level models. Challenges include finding a unit length capable of this and developing a selection method that results in both likeable and natural sounding music.

4.3 Reconstruction Using Unit Selection

As a first step towards evaluating the potential for unit selection, we examine how well a melody or a more complex jazz solo can be reconstructed using only the units available in a library. Two elements are needed to accomplish this: 1) data to build a unit library and 2) a method for analyzing a melody and identifying the best units to reconstruct it.

A dataset consisting of 4,235 lead sheets from the Wikifonia database containing melodies from genres including (but not limited to) jazz, folk, pop, and classical was used [181]. In addition, 120 publicly available jazz solo transcriptions from various websites were collected.

4.3.1 Design of a Deep Musical Autoencoder

In order to analyze and reconstruct a melody a deep autoencoder is trained to encode and decode a single measure of music. This means that the unit (in this scenario) is one measure of music. From the dataset there are roughly 170,000 unique measures. Of these, there are roughly 20,000 unique rhythms seen in the measures. The dataset is augmented by manipulating pitches through linear shifts (transpositions), tempo doubling, and alterations of the intervals between notes resulting in roughly 80 million unique measures. The interval alterations were performed in order to expand the dataset beyond what might be played by humans. For example, Shimon is better at playing fast sequences of notes that have large pitch intervals (greater than eight half steps) compared to small intervals. If these types of units are not present in the database then Shimon would never play them. By adjusting the intervals there is risk of sacrificing the musical validity because they are no longer human composed or confirmed. We attempt to maintain some validity by not modifying any of the rhythmic content. Furthermore, the number of units in the resulting library created by this process compose roughly 5% of the unit library. Therefore, the autoencoder

remains significantly biased towards effectively learning the semantics of the human valid units.

The intervals are altered using two methods: 1) adding a constant value to the original intervals and 2) multiplying a constant value to the intervals. Many different constant values are used and the resulting pitches from the new interval values are superimposed on to the measure's original rhythms. The new unit is added to the dataset. We restrict the library to measures with pitches that fall into a five octave range (midi notes 36-92). Each measure is transposed up and down continuously by half steps so that all instances within the pitch range are covered. The only manipulation performed on the duration values of notes within a measure is the temporal compression of two consecutive measures into a single measure. This "double time" representation effectively increases the number of measures, while leaving the inherent rhythmic structure in tact. After this manipulation and augmentation there are roughly 80 million unique measures. We use 60% for training and 40% for testing the autoencoder.

The first step in the process is feature extraction and creating a vector representation of the unit. Unit selection allows for a lossy representation of the events within a measure. As long as it is possible to rank the units it is not necessary to be able to recreate the exact sequence of notes with the autoencoder. Therefore, we can represent each measure using a bag-of-words (BOW) like feature vector. The features include:

1. counts of note tuples <pitch_1, duration_1>
2. counts of pitches <pitch_1>
3. counts of durations <duration_1>
4. counts of pitch class <class_1>
5. counts of class and rhythm tuples <class_1, duration_1>

6. counts of pitch bigrams <pitch_1, pitch_2>
7. counts of duration bigrams <duration_1, duration_2>
8. counts of pitch class bigrams <class_1, class_1>
9. first note is tied previous measure (1 or 0)
10. last note is tied to next measure (1 or 0)

These features were chosen in an attempt to keep the feature vector size to less than 10,000 so that training and computation could be performed in a reasonable time frame using a laptop computer. The pitches are represented using midi pitch values. The pitch class of a note is the note's pitch reduced down to a single octave (12 possible values). Rests of each duration are also included and treated similarly to pitched notes (like a 13th pitch class). We also represent rests using a pitch value equal to negative one. Therefore, no feature vector will consist of only zeros. Instead, if the measure is empty the feature vector will have a value of one at the position representing a whole rest. Because we used data that came from symbolic notation (not performance) the durations can be represented using their rational form (numerator, denominator) where a quarter note would be '1/4.' Finally, we also include beginning and end symbols to indicate whether the note is a first or last note in a measure. The resulting feature vector includes the counts and is not normalized, however, the unit size of one measure is constant.

The architecture of the autoencoder is depicted in **Figure 19**. The objective of the decoder is to reconstruct the feature vector and not the actual sequence of notes as depicted in the initial unit of music. Therefore, the entire process involves two types of reconstruction:

1. **feature vector reconstruction** - the reconstruction performed and learned by the *decoder*.

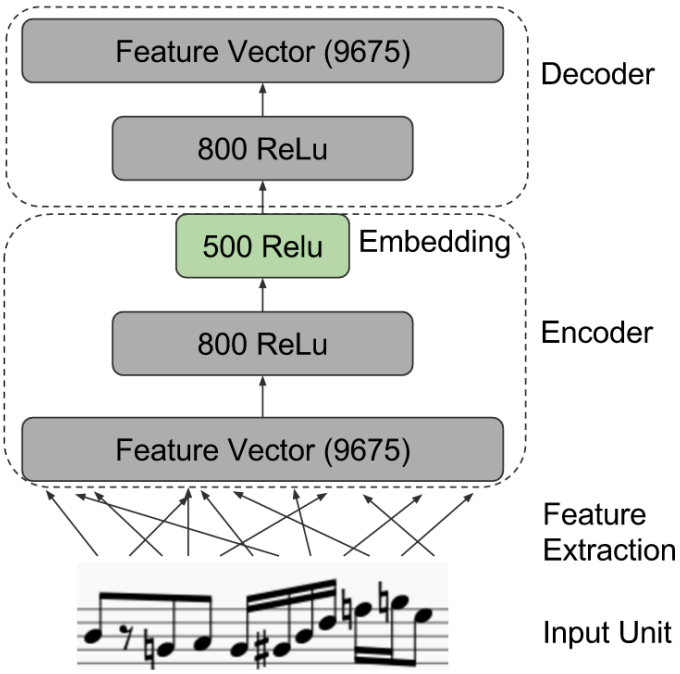


Figure 19: Autoencoder architecture – The unit is vectorized using a BOW like feature extraction and the autoencoder learns to reconstruct this feature vector.

2. **music reconstruction** - the process of selecting a unit that best represents the initial input musical unit.

In order for the network to learn the parameters necessary for effective feature vector reconstruction by the decoder, the network uses leaky rectified linear units (α=.001) on each layer and during training minimizes a loss function based on the cosine similarity function

$$sim(\tilde{X}, \tilde{Y}) = \frac{\tilde{X}^T \cdot \tilde{Y}}{|\tilde{X}||\tilde{Y}|} \tag{10}$$

where $\vec{X}$ and $\vec{Y}$ are two equal length vectors. This function serves as the basis for computing the distance between the input vector to the encoder and output vector of the decoder. Negative examples are included through a softmax function

$$P(\tilde{R}|\tilde{Q}) = \frac{\exp(sim(\tilde{Q}, \tilde{R}))}{\sum_{\tilde{d}\epsilon D} \exp(sim(\tilde{Q}, \tilde{d}))} \tag{11}$$

where $\vec{Q}$ is the feature vector derived from the input musical unit, Q, and $\vec{R}$ represents the reconstructed feature vector of Q. D is the set of five reconstructed feature vectors that includes $\vec{R}$ and four candidate reconstructed feature vectors derived from four randomly selected units in the training set. The network then minimizes the following differentiable loss function using gradient descent

$$-log \prod_{(Q,R)} P(\tilde{R}|\tilde{Q}) \tag{12}$$

A learning rate of 0.005 was used and a dropout of 0.5 was applied to each hidden layer, but not applied to the feature vector. The network was developed using Google's *Tensorflow* framework [1].

4.3.2 Music Reconstruction through Selection

The feature vector used as the input to the autoencoder is a BOW-like representation of the musical unit. This is not a loss-less representation and there is no effective means of converting this representation back into its original symbolic musical form. However, the nature of a unit selection method is such that it is not necessary to reconstruct the original sequence of notes. Instead, a candidate is selected from the library that best depicts the content of the original unit based on some distance metric.

In TTS, this distance metric is referred to as the *target cost* and describes the distance between a unit in the database and the target it's supposed to represent [223]. In this musical scenario, the targets are individual measures of music and the distance (or cost) is measured within the embedding space learned by the autoencoder. The unit whose embedding vector shares the highest cosine similarity with the query embedding is chosen as the top candidate to represent a query or target unit. We apply the function

$$\hat{y} = \arg\max_{y} sim(x, y) \tag{13}$$

where x is the embedding of the input unit and y is the embedding of a unit chosen

from the library.

The encoding and selection can be objectively and qualitatively evaluated. For the purposes of this particular musical autoencoder, an effective embedding is one that captures perceptually significant semantic properties and is capable of distinguishing the original unit in the library (low collision rate) despite the reduced dimensionality. In order to assess the second part we can complete a ranking (or sorting) task in which the selection rank (using equation 5) of the truth out of 49 randomly selected units (rank@50) is calculated for each unit in the test set. The collision rate can also be computed by counting the instances in which a particular embedding represents more than one unit. The results are reported Table 9.

Table 9: Autoencoder ranking and collision results

mean rank @ 50	1.003
accuracy @ 50	99.98
collision rate per 100k	91

Given the good performance we can make a strong assumption that if an identical unit to the one being encoded exists in the library then the reconstruction process will correctly select it as having the highest similarity. In practice, however, it is probable that such a unit will not exist in the library. The number of ways in which a measure can be filled with notes is insurmountably huge and the millions of measures in the current unit library represent only a tiny fraction of all possibilities. Therefore, in the instances in which an identical unit is unavailable an alternative, though perceptually similar, selection must be chosen.

Autoencoders and embeddings developed for image processing tasks are often qualitatively evaluated by examining the similarity between original and reconstructed images [200]. Likewise, we can assess the selection process by reconstructing never before seen music.

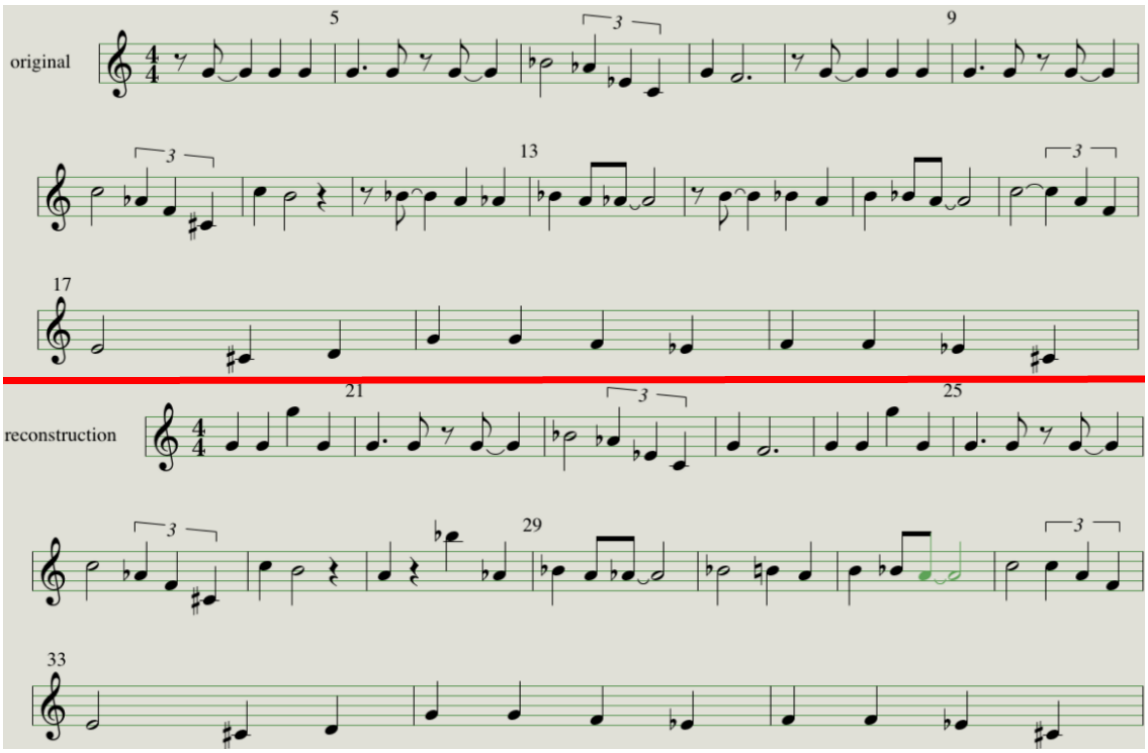


Figure 20: The music on the stave labeled "reconstruction" (below the line) is the reconstruction (using the encoding and unit selection process) of the music on the stave labeled "original" (above the line).

Figure 20 shows the reconstruction of an improvisation (see the related video for audio examples [1]). Through these types of reconstructions we are able to see and hear that the unit selection performs well. Also, note that this method of reconstruction utilizes only a target cost and does not include a concatenation cost between measures.

Another method of qualitative evaluation is to reconstruct from embeddings derived from linear interpolations between two input seeds. The premise is that the reconstruction from the vector representing the weighted sum of the two seed embeddings should result in samples that contain characteristics of both seed units. **Figure 21** shows results of reconstruction from three different pairs of units.

[1] `https://youtu.be/BbyvbO2F7ug`

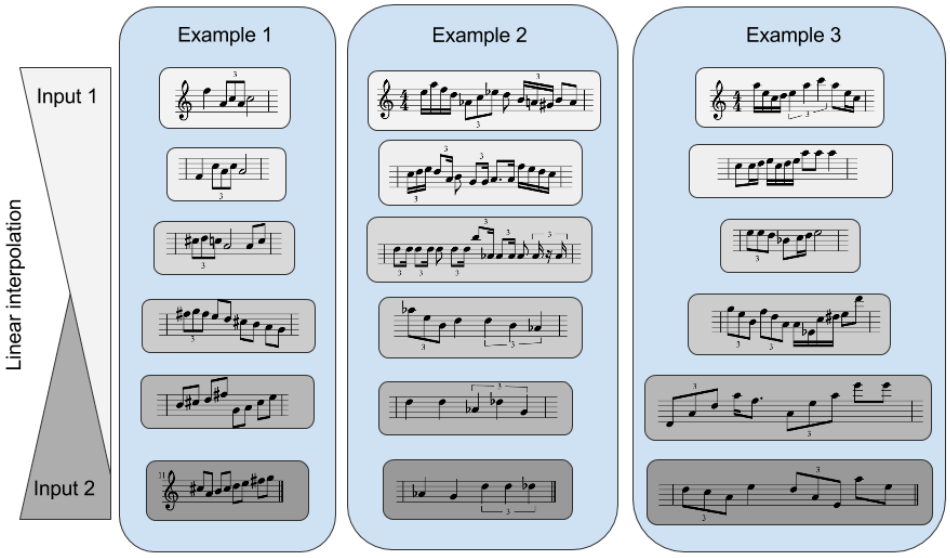


Figure 21: Linear interpolation in the embedding space in which the top and bottom units are used as endpoints in the interpolation. Units are selected based on their cosine similarity to the interpolated embedding vector.

4.4 Generation using Unit Selection

In the previous section we demonstrated how unit selection and an autoencoder can be used to transform an existing piece of music through reconstruction and merging processes. The embeddings learned by the autoencoder provide features that are used to select the unit in the library that best represents a given query unit. In this section we explore how unit selection can be used to generate sequences of music using a predictive method. The task of the system is to generate sequences by identifying good candidates in the library to contiguously follow a given unit or sequence of units.

The process for identifying good candidates is based on the assumption that two contiguous units, (u_{n-1}, u_n), should share characteristics in a higher level musical semantic space (semantic relevance) and the transition between the last and first notes of the first and second units respectively should be likely to occur according to a model (concatenation). This general idea is visually portrayed in **Figure 22**. We use a DSSM based on BOW-like features to model the semantic relevance between two

contiguous units and a note-level LSTM to learn likely note sequences (where a note contains pitch and rhythm information).

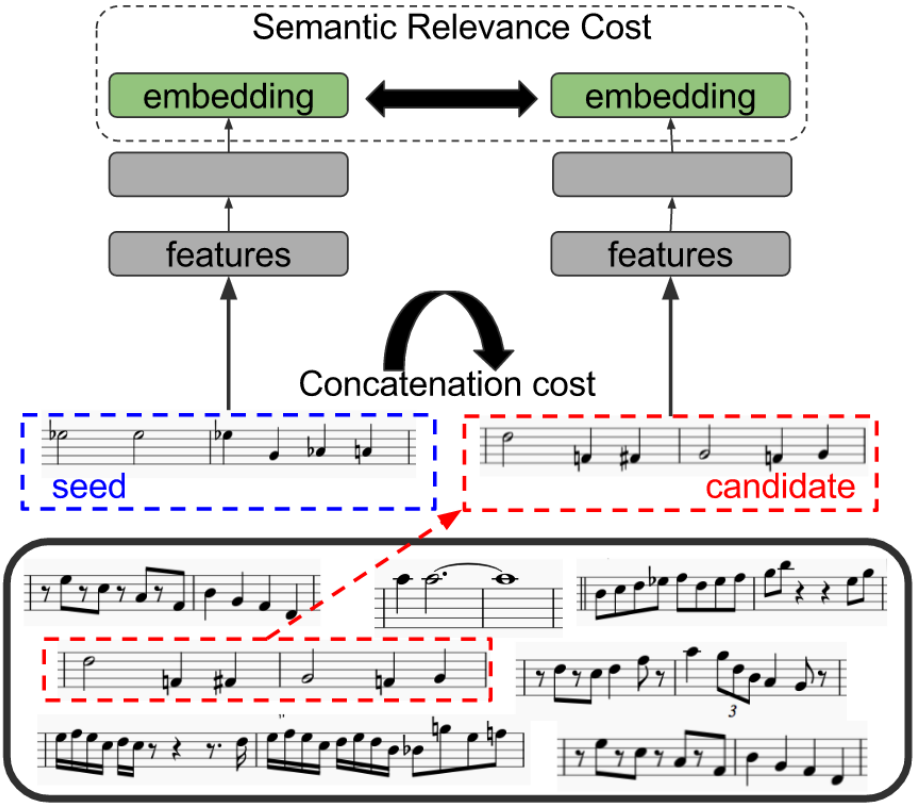


Figure 22: A candidate is picked from the unit library and evaluated based on a concatenation cost that describes the likelihood of the sequence of notes (based on a note-level LSTM) and a semantic relevance cost that describes the relationship between the two units in an embedding space (based on a DSSM).

For training these models we use the same dataset described in the previous section. However, in order to ensure that the model learns sequences and relationships that are musically appropriate we can only augment the dataset by transposing the pieces to different keys. Transposing does not compromise the original structure, pitch intervals, or rhythmic information within the data, however, the other transformations do affect these musical attributes and such transformations should not be applied for learning the parameters of these sequential models. However, it is possible to use the original unit library (including augmentations) when selecting units during generation.

4.4.1 Semantic Relevance

In both TTS and the previous musical reconstruction tests a target is provided. For generation tasks, however, the system must predict the next target based on the current sequential and contextual information that is available. In music, even if the content between two contiguous measures or phrases is different, there exist characteristics that suggest the two are not only related, but also likely to be adjacent to one another within the overall context of a musical score. We refer to this likelihood as the "semantic relevance" between two units.

This measure is obtained from a feature space learned using a DSSM. Though the underlying premise of the DSSM is similar to the autencoder in that the objective is to learn good features in a compressed semantic space, the DSSM features, however, are derived in order to describe the relevance between two different units by specifically maximizing the posterior probability of consecutive units, $P(u_n|u_{n-1})$, found in the training data. This idea stems from word in embeddings in which a word learns the context in which it would be used. The DSSM model and others such as the skip-gram have demonstrated to learn effective embeddings using this method [75, 129]. Recently, efficacy has been demonstrated in music using the skip-gram model on chord progressions. In this context, the embeddings learn a pitch space similar to that of the circle of fifths [85].

In this work, a space representing both rhythm and pitch features is learned. The same BOW features described in the previous section are used as input to the model. There are two hidden layers and the output layer describes the semantic feature vector used for computing the relevance. Each layer has 128 rectified linear units. The same softmax that was used for the autoencoder for computing loss is used for the DSSM. However, the loss is computed within vectors of the embedding space such that

$$-log \prod_{(u_{n-1},u_n)} P(\tilde{u_n}|\tilde{u_{n-1}}) \tag{14}$$

where the vectors, $\vec{u_n}$ and $\vec{u_{n-1}}$, represent the 128 length embeddings of each unit derived from the parameters of the DSSM. Once the parameters are learned through gradient descent the model can be used to measure the relevance between any two units, U_1 and U_2, using cosine similarity $sim(\tilde{U_1}, \tilde{U_2})$ (see Equation 1).

The DSSM provides a meaningful measure between two units, however, it does not describe how to join the units (which one should come first). Similarly, the BOW representation of the input vector does not contain information that is relevant for making decisions regarding sequence. In order to optimally join two units a second measure is necessary to describe the quality of the join.

4.4.2 Concatenation Cost

By using a unit library made up of original human compositions or improvisations, we can assume that the information within each unit is musically valid. In an attempt to ensure that the music remains valid after combining new units we employ a concatenation cost to describe the quality of the join between two units. This cost requires sequential information at a more fine grained level than the BOW-DSSM can provide.

The general premise for computing the quality of the concatenation is shown in **Figure 23**. A multi-layer LSTM is used to learn a note-to-note level model. This is akin to a character level language model. Each state in the model represents an individual note that is defined by its pitch and duration. This constitutes about a 3,000 note vocabulary. Using a one-hot encoding for the input, the model is trained to predict the next note, y_T, given a sequence, $\mathbf{x} = (x_1, ..., x_T)$, of previously seen notes. During training, the output sequence, $\mathbf{y} = (y_1, ..., y_T)$, of the network is such that $y_t = x_{t+1}$. Therefore, the predictive distribution of possible next notes, $Pr(x_{T+1}|\mathbf{x})$, is

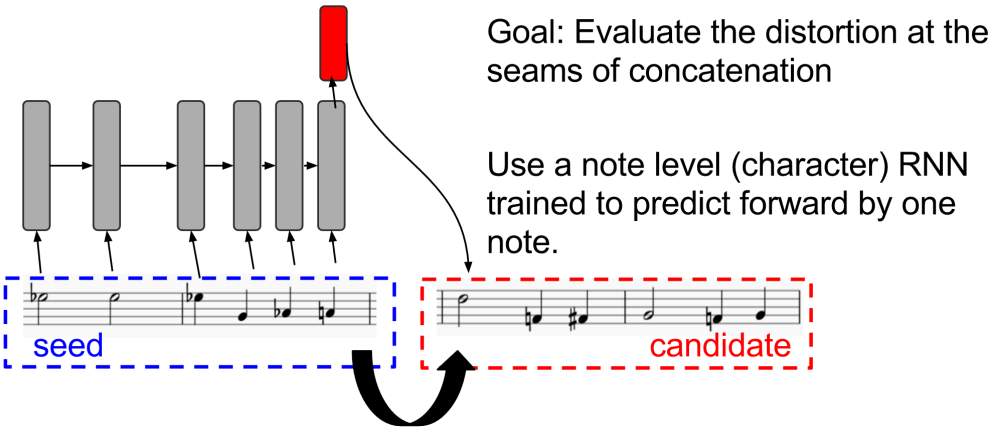


Figure 23: The concatenation cost is computed by evaluating the sequence of notes where two units join.

represented in the output vector, y_T. We use a sequence length of $T = 36$.

The aim of the concatenation cost is to compute a score evaluating the transition between the last note of the unit, u_{n-1,x_T}, and the first note of the unit, u_{n,y_T}. By using an LSTM it is possible to include additional context and note dependencies that exist further in the past than u_{n-1,x_T}. The cost between two units is computed as

$$C(u_{n-1}, u_n) = -\frac{1}{J}\sum_{j}^{J} logPr(x_j|\mathbf{x_j}) \tag{15}$$

where J is the number of notes in u_n, x_j is the jth note of u_n, and $\mathbf{x_j}$ is the sequence of notes (with length T) immediately before x_j. Thus, for $j > 1$ and $j < T$, $\mathbf{x_j}$ will include notes from u_n and u_{n-1} and for $j \geq T$, $\mathbf{x_j}$ will consist of notes entirely from u_n. In practice, however, the DSSM performs better than the note-level LSTM for predicting the next unit and we found that computing C with $J = 1$ provides the best performance. Therefore, the quality of the join is determined using only the first note of the unit in question (u_n).

The sequence length, $T = 36$, was chosen because it is roughly the average number of notes in four measures of music (from our dataset). Unlike the DSSM, which computes distances based on information from a fixed number of measures, the context

provided to the LSTM is fixed in the number of notes. This means it may look more or less than four measures into the past. In the scenario in which there is less than 36 notes of available context the sequence is zero padded.

4.4.3 Ranking Units

A ranking process that combines the semantic relevance and concatenation cost is used to perform unit selection. Often times in music generation systems the music is not generated deterministically, but instead uses a stochastic process and samples from a distribution that is provided by the model. One reason for this is that note-level Markov chains or LSTMs may get "stuck" repeating the same note(s). Adding randomness to the procedure helps to prevent this. Here, we describe a deterministic method as this system is not as prone to repetitive behaviors. However, it is simple to apply stochastic decision processes to this system as the variance provided by sampling can be desirable if the goal is to obtain many different musical outputs from a single input seed.

The ranking process is performed in four steps:

1. Rank all units according to their semantic relevance with an input seed using the feature space learned by the DSSM.

2. Take the units whose semantic relevance ranks them in the top 5% and re-rank based on their concatenation cost with the input.

3. Re-rank the same top 5% based on their combined semantic relevance and concatenation ranks.

4. Select the unit with the highest combined rank.

By limiting the combined rank score to using only the top 5% we are creating a bias towards the semantic relevance. The decision to do this was motivated by findings from pilot listening tests in which it was found that a coherent melodic sequence

Table 10: Unit Ranking

Model	Unit length (measures)	Acc	Mean Rank @ 50 + Standard Dev.
LSTM	4	17.2%	14.1 ±5.6
DSSM	4	33.2%	6.9 ±3.8
DSSM+LSTM	4	36.5%	5.9 ±2.7
LSTM	2	16.6%	14.8 ±5.8
DSSM	2	24.4%	10.3 ±4.3
DSSM+LSTM	2	28.0%	9.1 ±3.8
LSTM	1	16.1%	15.7 ±6.6
DSSM	1	19.7%	16.3 ±6.6
DSSM+LSTM	1	20.6%	13.9 ±4.1

relies more on the stylistic or semantic relatedness between two units than a smooth transition at the point of connection.

4.4.4 Evaluating the model

The model's ability to choose musically appropriate units can be evaluated using a ranking test. The task for the model is to predict the next unit given a never before seen four measures of music (from the held out test set). The prediction is made by ranking 50 candidates in which one is the truth and the other 49 are units randomly selected from the database. We repeat the experiments for musical units of different lengths including four, two, and one measures. The results are reported in Table 10 and they are based on the concatenation cost alone (LSTM), semantic relevance (DSSM), and the combined concatenation and semantic relevance using the selection process described above (DSSM+LSTM). The accuracy depicts the rate at which the truth is ranked the best.

4.4.5 Discussion

The primary benefit of unit selection is being able to directly apply previously composed music. The challenge is stitching together units such that the musical results are

stylistically appropriate and coherent. Another challenge in building unit selection systems is determining the optimal length of the unit. The goal is to use what has been seen before, yet have flexibility in what the system is capable of generating. The results of the ranking task may indicate that units of four measures have the best performance, yet these results do not provide any information describing the quality of the generated music.

Music inherently has a very high variance (especially when considering multiple genres). It may be that unit selection is too constraining and note-level control is necessary to create likeable music. Conversely, it may be that unit selection is sufficient and given an input sequence there may be multiple candidates within the unit database that are suitable for extending the sequence. In instances in which the ranking did not place the truth with the highest rank, we cannot assume that the selection is "wrong" because it may still be musically or stylistically valid. Given that the accuracies are not particularly high in the previous task, an additional evaluation step is necessary to both evaluate the unit lengths and to confirm that the decisions made in selecting units are musically appropriate. In order to do this a subjective listening test is necessary.

4.5 Subjective Evaluation

The mean rank metric of the last task provides a rough estimate that the model has learned something of importance. However, to be certain that the space is perceptually and musically meaningful, a user study is necessary. In this section a subjective listening test is described. Participants included 32 music experts in which a music expert is defined as an individual that has or is pursuing a higher level degree in music, a professional musician, or a music educator. Four systems were evaluated. Three of the systems employed unit selection using the DSSM+LSTM approach with unit lengths of four, two, and one measures. The fourth system used the note-level LSTM

to generate each note at a time.

The design of the test was inspired by subjective evaluations used by the TTS community. To create a sample each of the four systems was provided with the same input seed (retrieved from the held out dataset) and from this seed each system then generated four additional measures of music. This process results in four eight-measure music sequences with the same first four measures. The process was repeated 60 times using random four measure input seeds.

In TTS evaluations participants are asked to rate the quality of the synthesis based on naturalness and intelligibility [189]. In music performance systems the quality is typically evaluated using naturalness and likeability [106]. For a given listening sample, a participant is asked to listen to four eight-measure sequences (one for each system) and then are asked to rank the candidates within the sample according to questions pertaining to:

1. Naturalness of the transition between the first and second four measures.
2. Stylistic relatedness of the first and second four measures.
3. Naturalness of the last four measures.
4. Likeability of the last four measures.
5. Likeability of the entire eight measures.

Each participant was asked to evaluate 10 samples that were randomly selected from the original 60, thus, all participants listened to music generated by the same four systems, but the actual musical content and order randomly differed from participant to participant. The tests were completed online with an average duration of roughly 80 minutes.

4.5.1 Results

Rank order tests provide ordinal data that emphasize the relative differences among the systems. The average rank was computed across all participants similarly to TTS-MOS tests. The percent of being top ranked was also computed. These are shown in **Figures 24 and 25**.

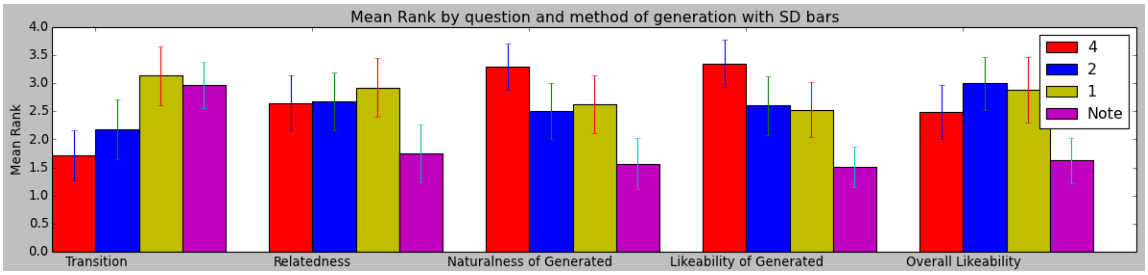


Figure 24: The mean rank and standard deviation for the different music generation systems using units of lengths 4, 2, and 1 measures and note level generation. A higher mean rank indicates a higher preference (i.e. higher is better).

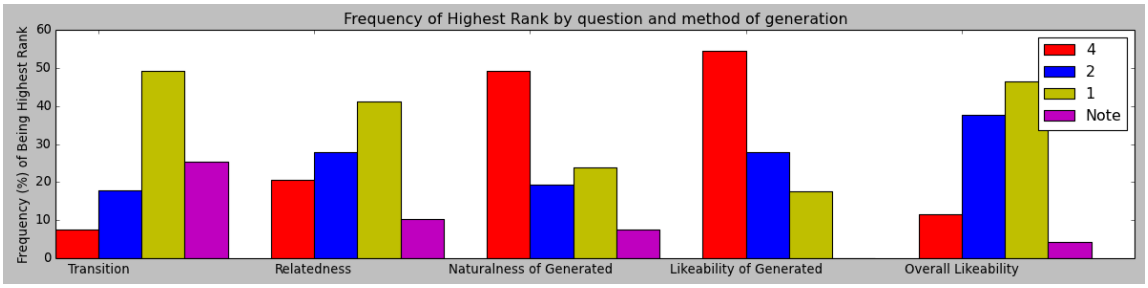


Figure 25: The frequency of being top ranked for the different music generation systems using units of lengths 4, 2, and 1 measures and note level generation. In both Figure 5 and 6 results are reported for each of the five hypotheses: 1) **Transition** – the naturalness of the transition between the first four measures (input seed) and last four measures (computer generated), 2) **Relatedness** – the stylistic or semantic relatedness between the first four measures and last four measures, 3) **Naturalness of Generated** – the naturalness of the last four measures only, 4) **Likeability of Generated** – the likeability of the last four measures only, and 5) **Overall Likeability** – the overall likeability of the entire eight measure sequence.

In order to test significance the non-parametric Friedman test for repeated measurements was used. The test evaluates the consistency of measurements (ranks) obtained in different ways (audio samples with varying input seeds). The null hypothesis states

Table 11: Subjective Ranking

Variable	Best –>Worst
H1 - Transition Naturalness	1, N, 2, 4
H2 - Semantic Relatedness	1, 2, 4, N
H3 - Naturalness of Generated	4, 1, 2, N
H4 - Likeability of Generated	4, 2, 1, N
H5 - Overall Likeability	2, 1, 4, N

that random sampling would result in sums of the ranks for each music system similar to what is observed in the experiment. A bonferonni post-hoc correction was used to correct the p-value for the five hypotheses (derived from the itemized question list described earlier).

For each hypothesis the Friedman test resulted in $p<.05$, thus, rejecting the null hypothesis. The sorted ranks for each of the generation system is described in Table 11.

4.5.2 Discussion

In H3 and H4 the participants were asked to evaluate the quality of the four generated measures alone (disregarding the seed). This means that the sequence resulting from the system that generates units of four measure durations are the unadulterated four measure segments that occurred in the original music. Given there was no computer generation or modification it is not surprising that the four measure system was ranked highest.

The note level generation performed well when it comes to evaluating the naturalness of the transition at the seams between the input seed and computer generated music. However, note level generation does not rank highly in the other categories. Our theory is that as the note-level LSTM accumulates error and gets further away from the original input seed the musical quality suffers. This behavior is greatly attenuated in a unit selection method assuming the units are pulled from human

compositions.

The results indicate that there exists an optimal unit length that is greater than a single note and less than four measures. This ideal unit length appears to be one or two measures with a bias seemingly favoring one measure. However, to say for certain an additional study is necessary that can better narrow the difference between these two systems.

4.6 An Embodied Unit Selection Process

So far this chapter has described a successful method of learning musical semantics and a method for generating music using unit selection. The selection process incorporates a score based on the semantic relevance between two units and a score based on the quality of the join at the point of concatenation. Two variables essential to the quality of the system are the breadth and size of the unit database and the unit length. An autoencoder was used to demonstrate the ability to reconstruct never before seen music by picking units out of a database. In the situation that an exact unit is not available the nearest neighbor computed within the embedded vector space is chosen. A subjective listening test was performed in order to evaluate the generated music using different unit durations. Music generated using units of one or two measure durations tended to be ranked higher according to naturalness and likeability than units of four measures or note-level generation.

Applying this generation system in its current form in the context of robotic musicianship would result in the same processing pipeline of *music generation→path planning→musical output* that this thesis is looking to avoid. Additionally, this system does not address situations in which the melodies should conform to a provided harmonic context (chord progression), therefore, it is not yet suitable for harmony-based jazz improvisation. This section addresses both of these issues.

4.6.1 Refining the selection process

In order to include traits of embodiment into the musicianship portion of this system the unit selection process should integrate variables describing the physical constraints. Currently, selections are made using two attributes describing the musical quality of joining two units (semantic relevance and concatenation cost). Though this information remains highly relevant, by itself it is not suitable for an embodied musical processing system that makes decisions by jointly optimizing for musical heuristics and physicality.

An additional cost, the *embodiment cost*, is computed describing the physical capability of playing a single unit. Using the path planning method described in the previous measure it is possible to determine if a robot can play a specific unit given its physical constraints and current state in the C-Space. This means that the robot's most recent configuration is vital for evaluating potential next units. In this case, the observation sequence in the Viterbi process comes from the notes sequence of a unit. The embodied cost is measured by performing Viterbi with this observation sequence and computing the most efficient movement sequence. If N represents the number of notes in the observation sequence O and T represents the number of notes that are performed given the Viterbi computed path for O then the embodiment cost, E, is the fraction of notes in the unit that can be played:

$$E = T/N \tag{16}$$

The goal is to pick a unit that the robotic system is fully capable of performing. Therefore, in this implementation all units with $E < 1.0$ are removed from consideration. The final unit selection procedure includes semantic relevance, concatenation cost, and embodiment cost. Using these metrics the robot creates musical opportunities based off of its current physical configuration and the learned musical heuristics, hence, satisfying Vijay Iyer's declaration that a good musician "requires an awareness of the palette of musical acts available in general, and particularly of the dynamically

evolving subset of this palette that is physically possible at any given moment" [91].

Figure 26 shows three generated sequences of music using the unit selection process. The first measure in each four measure sequence is the same and used as the seed to generate the next three measures. Examples are shown using unit selection with and without an embodiment measure.

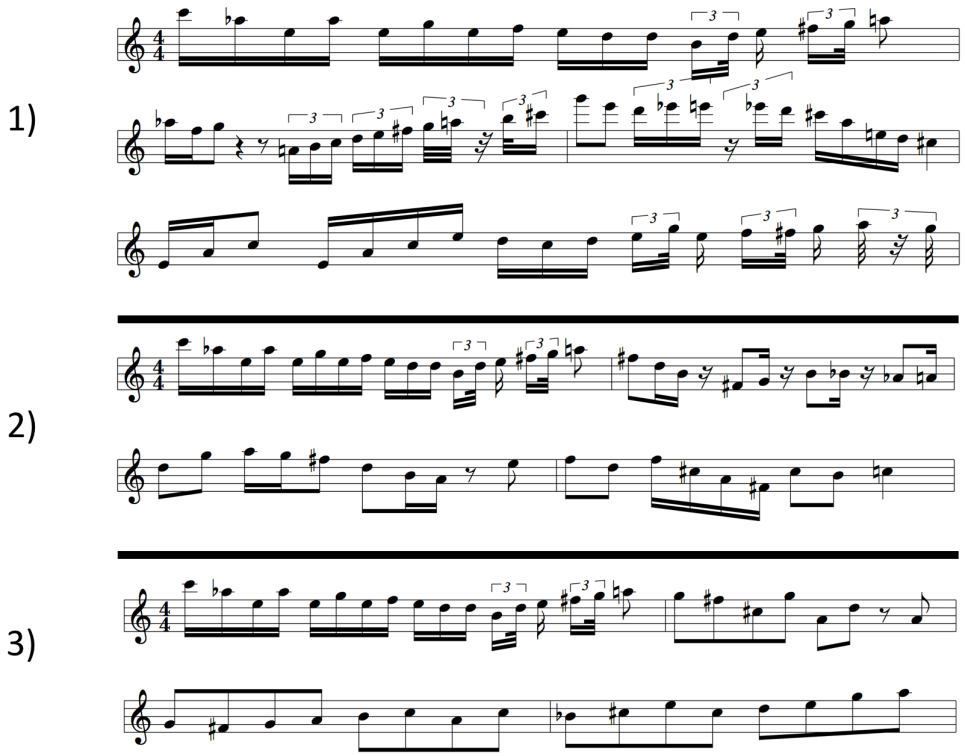


Figure 26: Three measures of music are generated using the unit selection process. The first measure in each sequence serves as the seed. Units are chosen using three different methodologies: (1) The units are selected using the semantic relevance and concatenation cost; (2) The units are selected using semantic relevance, concatenation cost, and an embodiment cost computed for the physical constraints of the Shimon robot; (3) The units are selected using semantic relevance, concatenation cost, and an embodiment cost based on a robot similar to Shimon, but with more significant speed limitations.

Unfortunately, an adequate dataset with labeled chord progressions and improvisations does not exist. Therefore, in order to use this system with a given chord progression a "note tweaking" step is applied. The process is fairly straightforward:

1. **Rank units according to musical quality** — The units are initially ranked

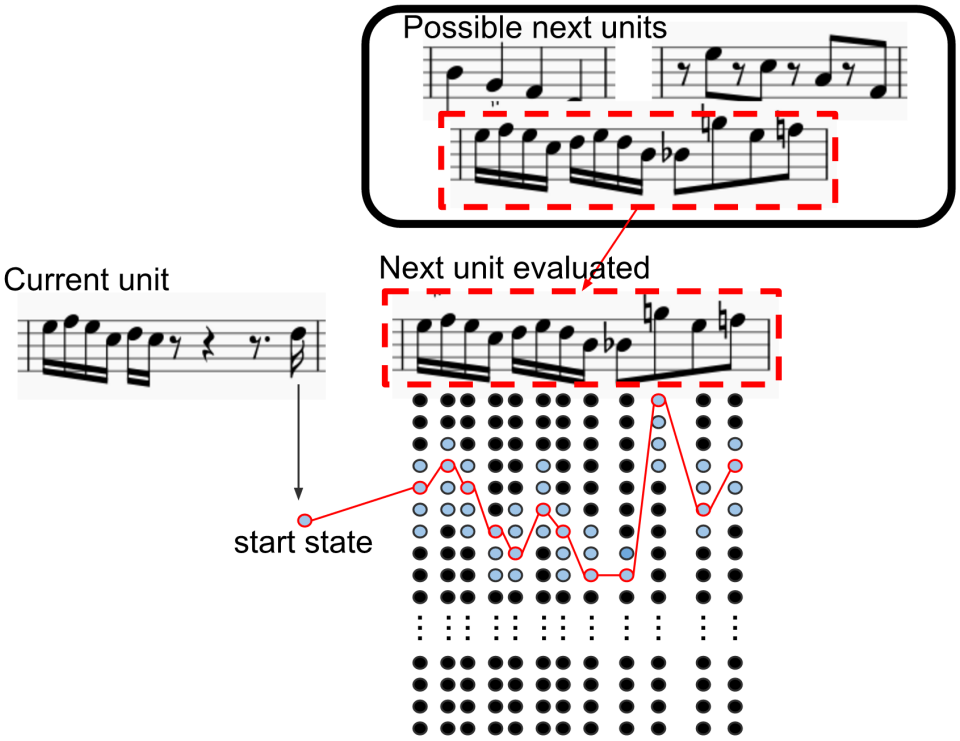


Figure 27: Once a unit is selected path planning is performed. The objective is to find a path that maintains the general contour and rhythm qualities of the unit, while finding a sequence that modifies the pitches such that they support the particular tonal center and chord function. Therefore, the search space is pruned to include only pitches close to each pitch in the unit. The generated sequence also addresses the physical constraints of the robot so the resulting score of the path describes both the unit's ability to appropriately fit the chord progression and robot's ability to play the notes. Each possible unit is evaluated according to this metric and the unit with the best score is chosen.

according to the semantic relevance and concatenation cost.

2. **Re-rank the top n units according to physicality and chord changes** — After units are initially sorted the top n units are selected for further evaluation based on the path planning algorithm.

Item two in this process has two objectives: 1) tweak pitches in the unit to fit a specific harmonic context and 2) find a movement path that jointly addresses the physical constraints of the robot. These objectives are integrated into a single task such that the result jointly optimizes across both objectives. The path planning

process is similar to the Viterbi planning of precomposed melodies (as opposed to the knowledge-based generative system), however, the pitches can be modified slightly. Therefore, the unit serves as a type of heuristic to help prune away possible branches in the search. In this implementation, the bias is fairly extreme and only allows the pitches of a unit to be modified by ±2 half steps. The pitches are modified according to their tonal distance from the observed chord (using the metric described in the previous chapter) and the robotic movement variables. The general pitch contour is kept and rhythm is maintained exactly.

$$\hat{U} = \arg\max_{U} PATH(x, U) \tag{17}$$

where x is the last state (in the robot and music c-space) of the current unit, Y is all the notes of a single unit chosen from the initially top n ranked units, and $PATH$ is a function denoting the resulting score of the path generated from the Viterbi decoding process. The process is outlined in **Figure 27**.

Ideally, units wouldn't be ranked and re-ranked in two separate processes. In practice, however, the Viterbi process is a computationally expensive metric, particularly if the state space is massive. An additional preprocessing step can be performed to reduce this expense. For each unit in the library, path planning can be performed. All units that are incapable of being performed in their current form (without pitch tweaking) can be removed from the database.

4.7 Conclusion

This chapter first described a method to learn musical semantics using deep learning. Next, the musical embeddings were applied to a generative music system based on unit selection and evaluated with a user study. Finally, a method for incorporating both physical constraints and harmonic context into the unit ranking process was described.

CHAPTER V

EMBODIED MUSICAL LEARNING

For an embodied system, learning the musical heuristics that should govern note choices is only one part of the equation. The theory and evidence for associative learning processes also imply that the physical parameters play a vital role in decision-making and perception. In this chapter, a method for learning the physical parameters simultaneously with the musical heuristics is described. The content of this chapter is largely exploratory and though there are many challenges to consider, this initial approach demonstrates the utility of addressing the physical constraints in the learning process.

5.1 Introduction

The previous chapter provided a good method for numerically describing a sequence of notes (the unit embedding), however, the musical heuristic learning and path planning were separate processes. Though the generation was influenced by the physical constraints of the robot, how the system perceives and interprets a musical motif is irrespective of any physicality or properties of embodiment. The model's embeddings are purely musical in nature. While this is quite useful and helps optimize decision-making, the theory of associative learning suggests that a human musician's perception and interpretation of music is influenced by his or her embodiment and the associations developed over time by learning an instrument [13, 54]. In a fully embodied cognitive system the embeddings should encapsulate information about the music as well as the body.

In this chapter, the concept of associative learning is explored as a computational task and seeks to address whether the musical as well as the physical heuristics that

govern musical path planning can be learned together. The approach proposed here moves away from unit selection and into a methodology in which all notes are generated. One caveat of unit selection is that the system is bound to the capacity and richness of the unit database. In situations in which a robot is incapable of performing the majority of the units this is not very useful. Despite the departure from unit selection, the proposed method draws from the success of the deep structured semantic model (DSSM) in the previous chapter and the effective embeddings learned that describe an entire measure. Unlike the typical generative systems which learn to predict one note at a time, this method generates an entire sequence all at once. In robotic musicianship the disadvantages of a note to note generative system is that it could fall into a physical configuration that may prevent it from performing adequately in the future. Planning prevents such situations and the potential repercussions. The objective, here, is to use a method that allows for note level flexibility while maintaining the benefits of planning and leveraging metrics that semantically describe entire sequences of notes.

5.2 Embodied Learning

5.2.1 Convolutional Autoencoder

Previously in this work, music has been represented using a score-like symbolic representation and network inputs were constructed from features derived from this representation. Here, a piano roll representation is used. This is because the network needs the previous feature extraction was lossy and could not be used to generate music. The goal in this section is to generate all of the necessary music structure (no unit selection), therefore, a representation must be used that allows for this.

In a piano roll representation a single measure of music can be thought of as a two-dimensional matrix with time in one axis and pitch in the other. The input to the model is a single measure (four beats) of music represented in this matrix format. However, only onsets are considered and the tick resolution is 24 ticks per beat.

As a first step, to demonstrate note generation, a convolutional autoencoder that does not consider embodiment or physical constraints is developed. An autoencoder takes an input $\mathbf{x} \in R^d$, maps it to the latent representation $\mathbf{h} \in R^d$ using $\mathbf{h} = f_\theta = \sigma(Wx + b)$ where σ is the activation function, and reconstructs the input from the latent space using the inverse mapping $f'_\theta(h) = \sigma(W'h + b')$. The network is trained to optimize the parameters $\theta = \{W, b\}$ by minimizing a given distance function. In order to capture local features that may repeat themselves convolution is used so that the latent representation of the k-th feature map becomes $h^k = \sigma(W^k + b^k)$.

To avoid simply learning the identify function the model learns to *denoise* a corrupted version of the input. In this case, 50% of the notes are zeroed out and trained to be reconstructed. The distance metric is a softmax function over cosine similarity between the output and the original (non-corrupted) input:

$$P(\tilde{R}|\tilde{Q}) = \frac{\exp(sim(\tilde{Q}, \tilde{R}))}{\sum_{\tilde{d}\epsilon D} \exp(sim(\tilde{Q}, \tilde{d}))} \tag{18}$$

where $\vec{R}$ is the reconstructed vector and $\vec{Q}$ is the original input.

The architecture of the network is depicted in **Figure 28**. The first layer convolves a 12×12 feature map (one octave of pitches and an eighth note duration worth of ticks) and has a stride rate of the same size. The subsequent convolutional layers use feature maps with stride rates that are half the size. Each layer uses exponential linear units (elu) and batch normalization is performed on each layer. The batch size is 100. The parameters of the encoder, $\theta = \{W, b\}$ and decoder, $\theta' = \{W', b'\}$ are constrained such that $W' = W^T$. A fully connected layer is used for the last layer of the encoder such that the embedding is a one-dimensional vector describing the entire measure.

The objective of the autoencoder is to reconstruct the notes perfectly. However, in practice, there is information loss and the output is not a perfect representation. Instead, a distribution of values over the piano roll space is generated. In this case, after training, the output is a distribution that exhibits a strong bias towards the

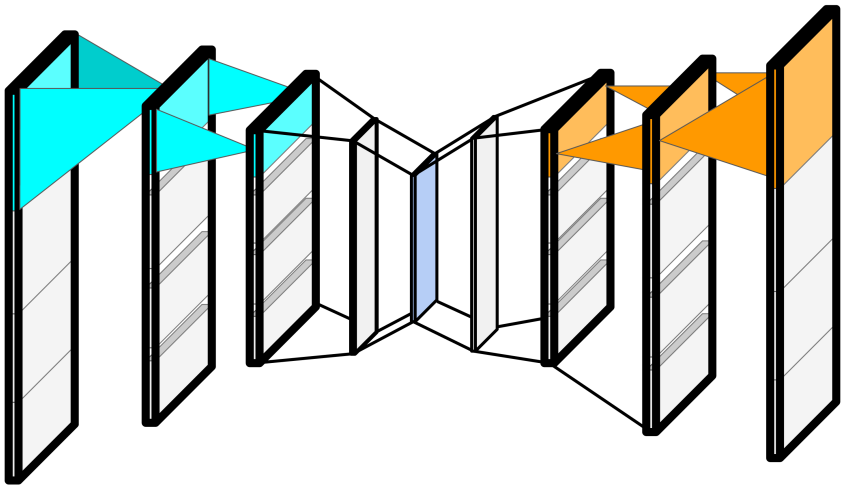


Figure 28: A denoising convolutional autoencoder is used to encode a piano roll representation of music. The input is a 60×96 matrix consisting of four beats of music (24 ticks per beat) and 5 octaves worth of pitches. The first two hidden layers convolutional and the third and fourth are full connected. The parameters of the encoder, $\theta = \{W, b\}$ and decoder, $\theta' = \{W', b'\}$ are constrained such that $W = W^T$.

notes of the original input with additional activated pitches. Playing back all the positive activations (even with activation strength mapped to volume) is not a good strategy and tends to sound poor. Instead, some decision process needs to occur that selects the notes that should be played **Figures 29 and 30**.

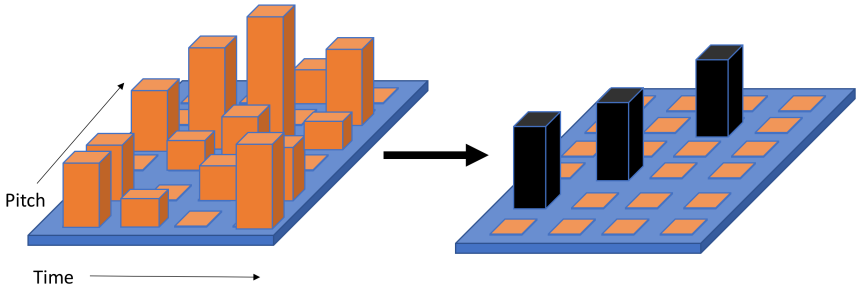


Figure 29: A method of choosing notes from the autoencoder output is necessary because the network will produce positive activations for a large number of notes.

The easiest method for choosing notes is to apply a pre-defined threshold and select all the notes with activations that exceed it. Adjusting this threshold value is related to modifying the temperature in a softmax sampling function. By decreasing the temperature the network produces results resembling hardmax decision-processes

and by increasing the temperature the differences between potential note candidates become less extreme (or softer). Typically, in music it is desirable to have the variation provided by a softmax output, but have the resulting music sound as if hardmax decisions were made. The general technique for achieving this is stochastic sampling. For a given time stamp in a measure, the model generates a distribution across all possible pitches and x pitches are chosen by performing a random weighted sampling of the distribution. These x pitches are then chosen to be played.

This type of decision process has potential for generating interesting music and some even argue that such processes are the basis for human creativity. Johnson-Laird [96] and Weinberg [206] argue that creative processes stem from procedural processes in which decisions are governed by chance and randomness. However, given the evidence for associative learning in music [72, 73], findings from Norgaard [152], and comments from the likes of Vijay Iyer [91], it is likely that this is not the case. The note-based musical decisions are probably not derived from random chance, but instead born out of an embodied system's motor-musical associations developed over time from a combination of musical listening, viewing, and practice. Furthermore, despite the true nature of creativity a selection process based on randomness may be sufficient for pure software applications, but is not ideal for a robotic musician because the process does not address the physical parameters of the system.

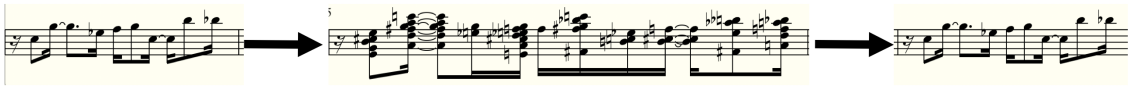


Figure 30: An input to the autoencoder is provided and the trained autoencoder reconstructs this input. The left measure represents the input. The middle measure shows all notes that have positive activations (using an exponential linear activation function). The right measure shows the result of a note selection method that chooses all notes that are at least seven standard deviations away from the mean.

5.2.2 Incorporating the Physical Constraints

Though learning to generate note sequences that automatically adhere to the physical constraints may more strongly mimic human associative learning tendencies, there is value beyond this. In the Viterbi decoding process the features navigating the path are extremely local, either one note (emission) or a transition between two notes (transition). However, higher level musical semantics describe note sequences consisting of many notes. For Viterbi to effectively capture semantics such as this a look back of many notes would be required (high dimensional n-grams). However, this is not feasible for extremely large state spaces. If a network could learn to generate note sequences that a robot is capable of playing then sequence generation could be possible when the state space that would be too computationally expensive to perform as a search task.

In the previous chapter it was shown that the embedding learned by the DSSM captured perceptually meaningful features (as given by the listening generation study). Assuming that the embedding projects a measure of music into a semantically relevant musical space, this space can be used to train additional models. In order to do this two competing networks are trained to generate note sequences, given a selection process $S(x)$ defined by the physical parameters of an arbitrary robotic system, that maximizes the semantic similarity between the the DSSM embedding of the original note sequence and the DSSM embedding of the generated note sequence (**Figure 31**).

The parameters of the DSSM encoder are represented as $\theta = \{W, b\}$ such that the latent musical feature, $\mathbf{h}$, (or embedding) of an input $\mathbf{x} \in R^d$ to the DSSM is $\mathbf{h} = f_\theta(x) = \sigma(W g(x) + b)$ where $g(x)$ is the resulting vector from the feature extraction described in the previous chapter. The parameters of two additional networks are represented as $\omega = \{W_1, b_1\}$ and $\lambda = \{W_2, b_2\}$. The output of these networks are represented as $\mathbf{d} = f_\omega(x) = \sigma(W_1 x + b_1)$ and $\mathbf{r} = f_\lambda(x) = \sigma(W_2 x + b_2)$, respectively (notice there is no feature extraction from this stage). The input and

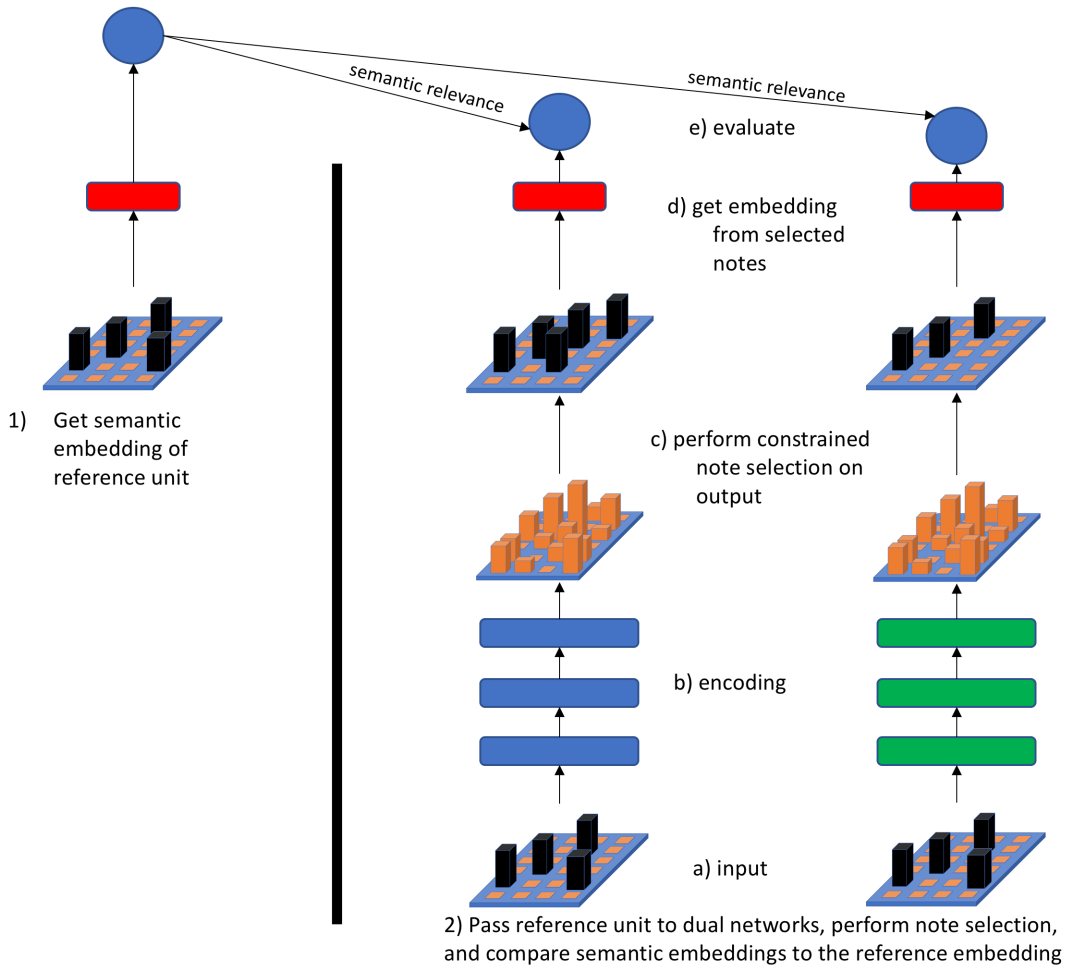


Figure 31: The training procedure designed to address varying physical parameters includes two competing networks. Notes are selected from the output of each network based on the physical parameters of the system. The semantic relevance (using the DSSM) is measured between the original input and the two resulting note sequences. The network that has the highest similarity in the musical semantic relevance space is the winner and the losing network is updated in the direction of the winning network.

outputs of these two networks have identical dimensions such that $\mathbf{x}, \mathbf{d}, \mathbf{r} \in R^{(m \times n)}$ where m is the number pitches and n is the number of midi ticks.

If $S(x)$ denotes the selection process of choosing notes from the distribution, the objective during training can be described as minimizing the distance between $f_\theta(S(\mathbf{d}))$ and $f_\theta(x)$ and between $f_\theta(S(\mathbf{r}))$ and $f_\theta(x)$. The DSSM embedding is used to project the selected notes into the learned musical space in order to measure the semantic relevance between the generated note sequence (as defined by the network and the

physically constrained selection process) and the original note sequence. The goal is for the networks to generate note sequences that adhere to the physical constraints of an embodied system, yet, are musically similar to the original sequence. After the semantic relevance is measured between the resulting two note sequences and the original note sequence a winner, ψ, is chosen (the result with the highest semantic relevance) and the losing network, ι, is updated in the direction of the winner using L_p loss.

$$D(\psi, \iota) = ||f_\psi(x) - f_\iota(x)||_p = (\sum_{i=1}^{n} |f_\psi(x)_i - f_\iota(x)_i|^p)^{1/p} \tag{19}$$

5.3 Evaluation

If the previously described training method functions as desired then the network output should prioritize notes that maximize their semantic relevance with an input seed. The following experiment evaluates the method's ability of achieving this.

5.3.1 Experiment

Consider a trumpet player that is given a piece of piano music and asked to play it such that he captures as much of the essence of the music as possible. Does he merely play the notes that are in the range of the trumpet? Does he transpose certain notes to the instrument's range? Which notes are dropped when there are chords? How is rhythm dealt with when there are interval jumps or melodic lines that are too fast to be played accurately on the trumpet? Presumably, the decision-making process that goes into this task integrates information regarding the affordances of the trumpet and knowledge about tonal harmony, transposition, and pitch relationships.

In this experiment, the utility of including the selection process based on physical constraints during training is measured. To do this, a hypothetical instrument is created with a set of affordances allowing it to play monophonically and only pitches

in a single octave, C4-C5 (a total of 13 notes beginning with middle 'C'). In the first system a convolutional autoencoder is trained as described previously with a single measure. The selection process consists of picking the pitch with the highest activation in the C4 octave for each tick that contains notes in the original input (such that resulting rhythm should be identical to the input). The output of the selection process can be thought of as a single octave and monophonic representation of the input.

The second system utilizes the method of training described for including physical constraints. The selection process is identical, however, dual networks are used and the losing network is updated in the direction of the winning network during training. After training and during execution, both networks are evaluated and the output with the highest semantic relevance is chosen as the output to be played.

The experiment includes a held out test of 1000 unique measures (polyphonic and monophonic) of music. The average cosine similarity, in the DSSM musical embedding space, between the result of the physically constrained networks and the pure autoencoder are measured.

5.3.2 Results

The average semantic relevance between an input seed and outputs of the autoencoder and model resulting from the constrained training process is reported in Table 12. The semantic relevance is computed using the trained DSSM described in the previous chapter. Samples from the generated outputs are shown in **Figure 32**.

Table 12: Results comparing average semantic relevance of original input to an autoencoder output and physically constrained network output. The average cosine similarity in the embedding space is reported.

network type	semantic relevance
autoencoder	.53
constrained	.66

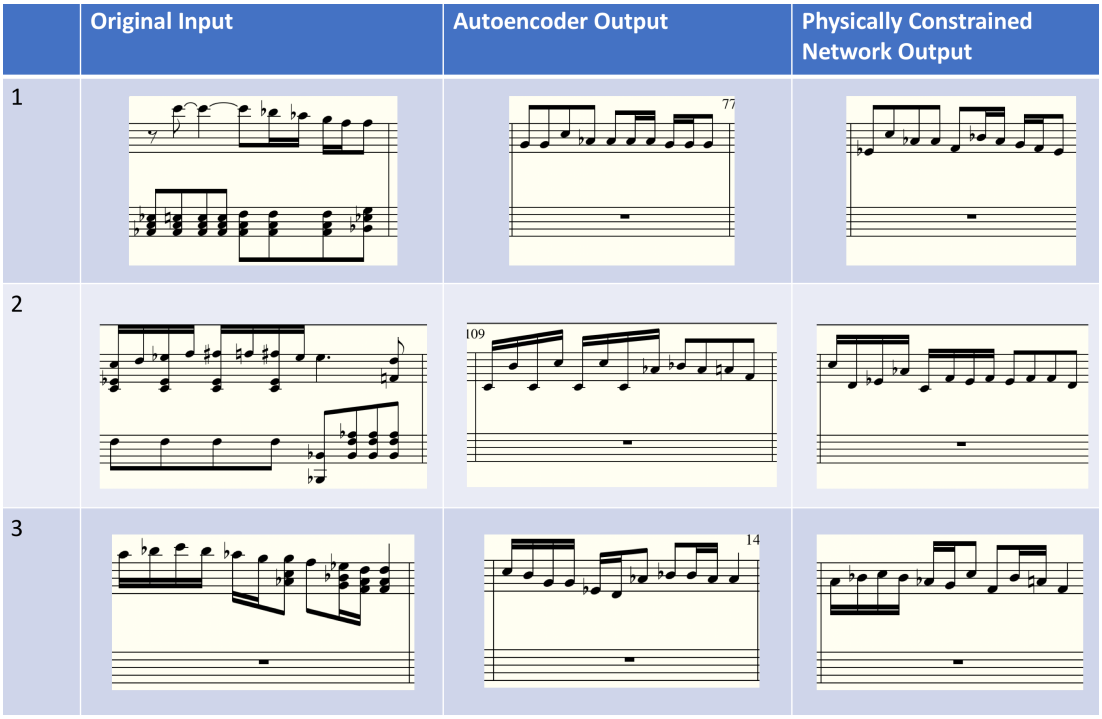


Figure 32: Three samples showing the original input, constrained sampling from the output of an autoencoder, and constrained sampling from the output of the network trained to incorporate physical parameters.

5.3.3 Discussion

The training process demonstrated efficacy in learning to generate outputs that would result in an improved semantic relevance score. Though this method of training demonstrates promise, there are still many challenges. The physical parameters in the experiment were relatively easy to implement and in practice a much more complex simulation and representation of the physical parameters of the system would need to be developed to capture all the subtleties and complexities of an actual embodied system.

Another method worth mentioning that was tried is to not implement any of the physical constraints, but rather just let the robotic system (Shimon in this case) play the sequence as best as possible in real-time. The semantic relevance was then measured on the resulting sequence. While this also showed promise in that the robot

was learning about itself in real-time while 'practicing' these motifs (much like in developmental robotic applications), the learning process takes much too long to be really useful. One thought for future work is to train the system as described here with a note selection process mimicking Shimon (or some other robot) as a pre-training step and then let the actual robot train using this real-time practicing methodology. Other modifications can be made by switching out the DSSM musical space with human rankings. The training process conveniently lends itself to a binary decision process (choosing one output as being better or worse) for a person to provide input in real-time. These methods bear great potential for future research and applications.

5.4 Conclusion

Addressing the affordances of an instrument during the compositional process seems obvious, but is a factor that has been ignored in autonomous music generation systems. When using machine learning techniques to acquire musicianship, the system by default develops priors or biases that tend towards the physical idiosyncrasies of whatever specific instrument is used in the dataset. By including the physical parameters in a 'note sampling' process it is able to generate more optimal note decisions based off of the system's embodiment.

This chapter described a training method that includes the physical parameters of a system. The method demonstrates efficacy compared to an autoencoder that learned without reference to the physical constraints. While the methodology is not perfect it bears promise for future work in music performance and generation by an embodied system.

CHAPTER VI

CONCLUSION

In this research an argument for robotic musicians to employ decision-making processes that jointly optimize for musical heuristics as well as their physical parameters is made. Several methods for how to achieve this are presented. In the following sections I revisit the specific contributions and briefly re-iterate the advantages and disadvantages of each.

6.1 Contributions

6.1.1 Path Planning

A Viterbi based method for playing precomposed note sequences is described. The system discretizes the physical and musical parameters into an integrated state space that enables a robot to generate efficient movement plans for performing the notes. Efficiency is measured on the robot's ability to avoid damage to itself (collisions), play all of the notes, avoid spurious movements, and conserve energy. The planning method shows improved efficiency compared to a greedy method.

A generative jazz improvisation system is designed on top of the Viterbi planning method. Instead of playing precomposed notes the pathfinding involves generating note sequences that simultaneously adhere to higher level musical semantics (tonal tension, pitch contour, note density, and rhythmic complexity) and the physical constraints of the system. By including the physicality into the decision-making process the system generates melodic sequences specifically designed for the physical configuration of an embodied system. This system is evaluated qualitatively and the resulting music demonstrates signs of non-human characteristics and emergence when the constraints allow for it.

The primary advantage of this method is the ability to find optimal paths. However, computational complexity of this task increases with the size of the state space. Regarding the generation, each musical semantic is manually designed. Though the semantics have demonstrated success in jazz improvisations and have been used in numerous performances they will likely not generalize to domains outside of jazz.

6.1.2 Learning Musical Semantics and Unit Selection

A method for learning higher-level musical semantics using deep learning is described. The semantics are learned using a deep structured semantic model (DSSM) that is trained to find the relevance between two adjacent musical units (where a unit is 1, 2, or 4 measures). The embeddings learned from this model are evaluated objectively using a ranking task. The embeddings are used in a novel concatenative synthesis system to generate sequences and further evaluated against against note-level generation using a subjective listening test. The physical parameters of a robotic system are included by re-ranking units based on the robot's ability to physically perform the unit's notes and tweak them to comply with tonal rules of jazz harmonic theory given a chord progression.

Unit selection methods lend themselves to the deep learning techniques used for discriminative tasks. Additionally, with unit selection the generative system gets some of the low level structure for free and the networks can focus their capacity on learning an effective latent space. Though effective embeddings were learning, the system's ability to generate music is restricted by the scope of the unit library. It is not guaranteed that the library will contain the necessary units for robots with varying physical constraints.

6.1.3 Learning the Physical Parameters

A method for automatic note generation using convolutional neural networks is described. First, a convolutional autoencoder is developed as a baseline. Next, a

method for training that modifies the weights according to the performance of a note selection process based on a the system's physical constraints is described. The system involves projecting the note sequences of two competing networks into the DSSM embedding space (from the unit selection model) and measuring distance within that space. The network that is most similar to the original output is declared the winner and the losing network is updated in the direction of the winning network. The methodology is compared to the autoencoding training process and demonstrates that including the physical parameters during training can help bias the network to create better note outputs (based on the semantic distance to the original input).

By learning to generate notes that are based on the physical constraints of a robot, the system frees itself from using only what is available in a unit library. However, generating all of the musical structure from the ground up is a difficult task and typically unit selection methods are qualitatively better.

6.2 Future Work

The groundwork laid by this work opens the doors for several avenues of research.

6.2.1 Integrating Social Cues

In this work the musical decisions have been shown that they can be influenced by the sound-producing robotic movements, however, this work does not address any potential influence by sound-accompanying movements. Conveying additional information (for the audience or interacting musicians) through the use of social cues can be musically beneficial. For example, attributes such as rhythmic structure, phrasing, and emotion can be conveyed through accompanying motions. Including these types of parameters to help shape the musical decisions will lead to more convincing performances by robot in which the system truly looks as if it is being expressive. It is not clear how best to represent these movements in the state space and the additional complexity will cause further challenges. However, they are worth tackling, particularly in the

domain of music in which expression and social interaction are integral to a successful performance.

6.2.2 Alternative Models for Embodied Learning

Some of the problem and challenges addressed in this work may lend themselves to alternative models. For example, the embodied learning method utilized a convolutional neural network. To address robotic systems in which the constraint effects the rhythmic outputs as well as the pitches, an LSTM can be useful. However, an LSTM should not be used to just predict the next note (as is typical), but instead generate entire sequences and leverage the benefits of planning. Therefore, techniques such as bi-directional LSTMs may be useful for this purpose.

6.2.3 Perception

The embeddings learned in this work may be thought of as perceptual tools as they are the system's internal representations of music. However, it may be useful to include additional features in this internal representation. Musicians (humans and robots alike) that interact with other musicians should understand the embodiment, the physical parameters, and instrument affordances of the interacting musicians in addition to their own. By understanding the capabilities one has with his or her instrument, a musical system can make more informed decisions about how to respond. For example, a future robot can use the bounds of attributes such as note density or pitch that are usually defined by the instrument and a person's ability to interact with it.

6.3 Final Remarks

In reading this dissertation I hope that you, the reader, are left with a newfound interest in why bodies matter, how we may use our bodies to think, and how a unique physical design of robots can and should influence their thinking. For scientists, I

hope you see the interesting questions (and I hope I answered a few of them) that are embedded within the realm of robotic musicianship and how we can build intelligent agents that utilize a form of embodied cognition. For musicians, I hope you see the potential for new types of creativity and musical styles that may emerge from robots that do not share the same physical constraints to which we are bound as humans.

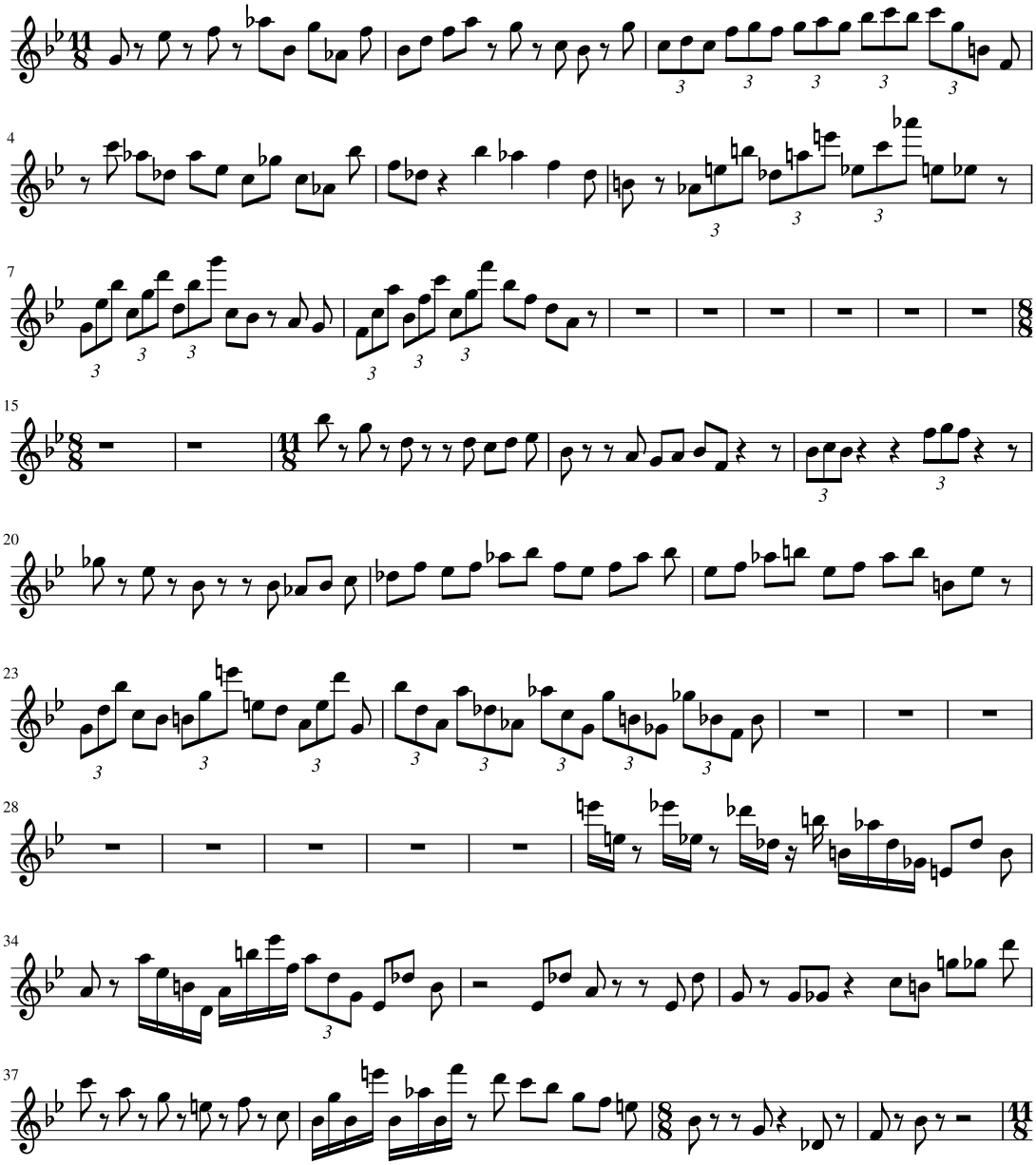
APPENDIX A

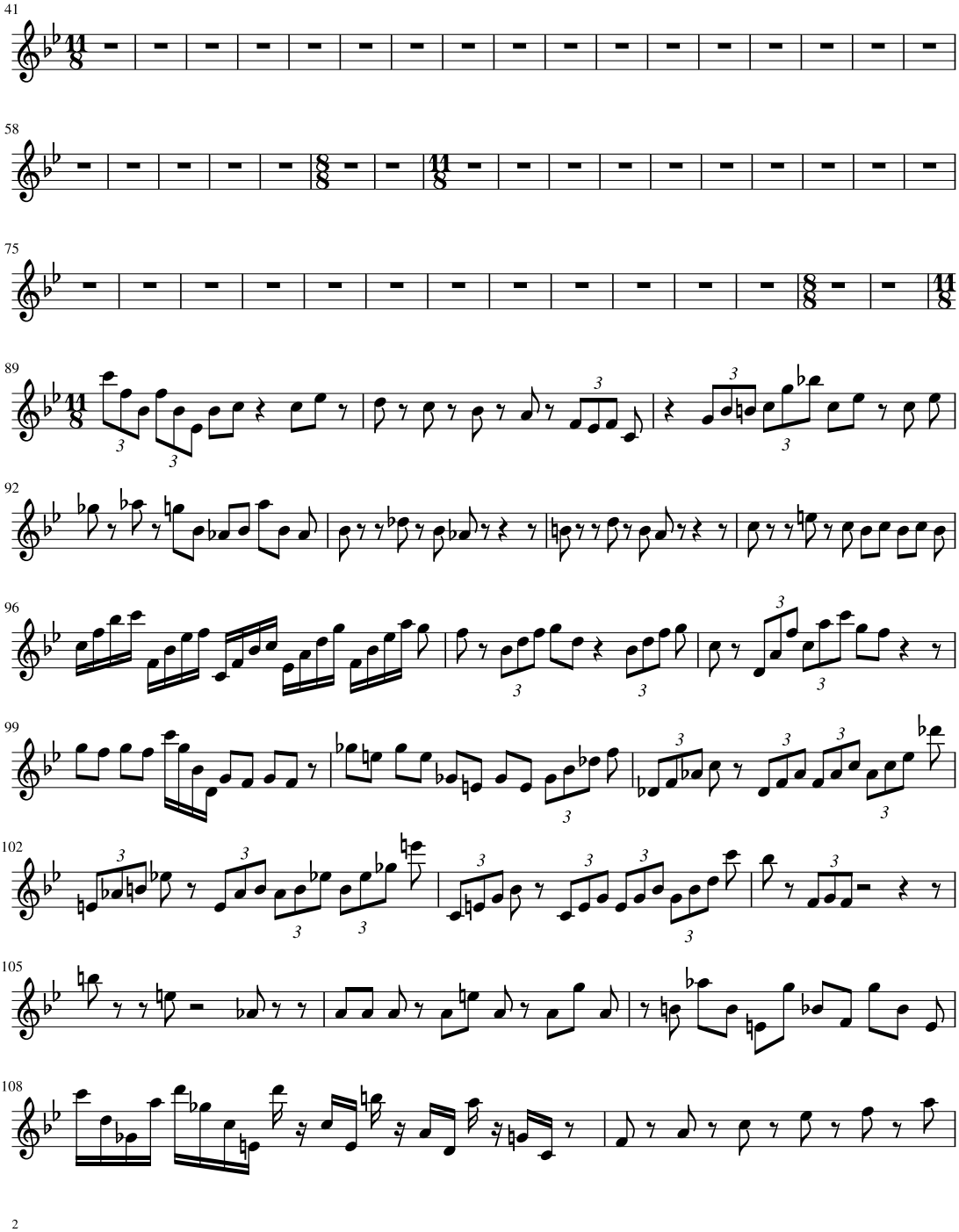
GENERATED MUSIC

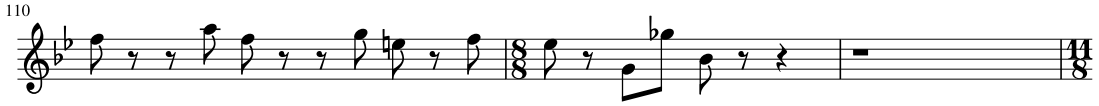
A.1 Shimon improvisation

Below is an improvisation generated by the path planning system using the set of physical constraints for the Shimon robot.

Iltur 2 - Shimon Generated Improvisation

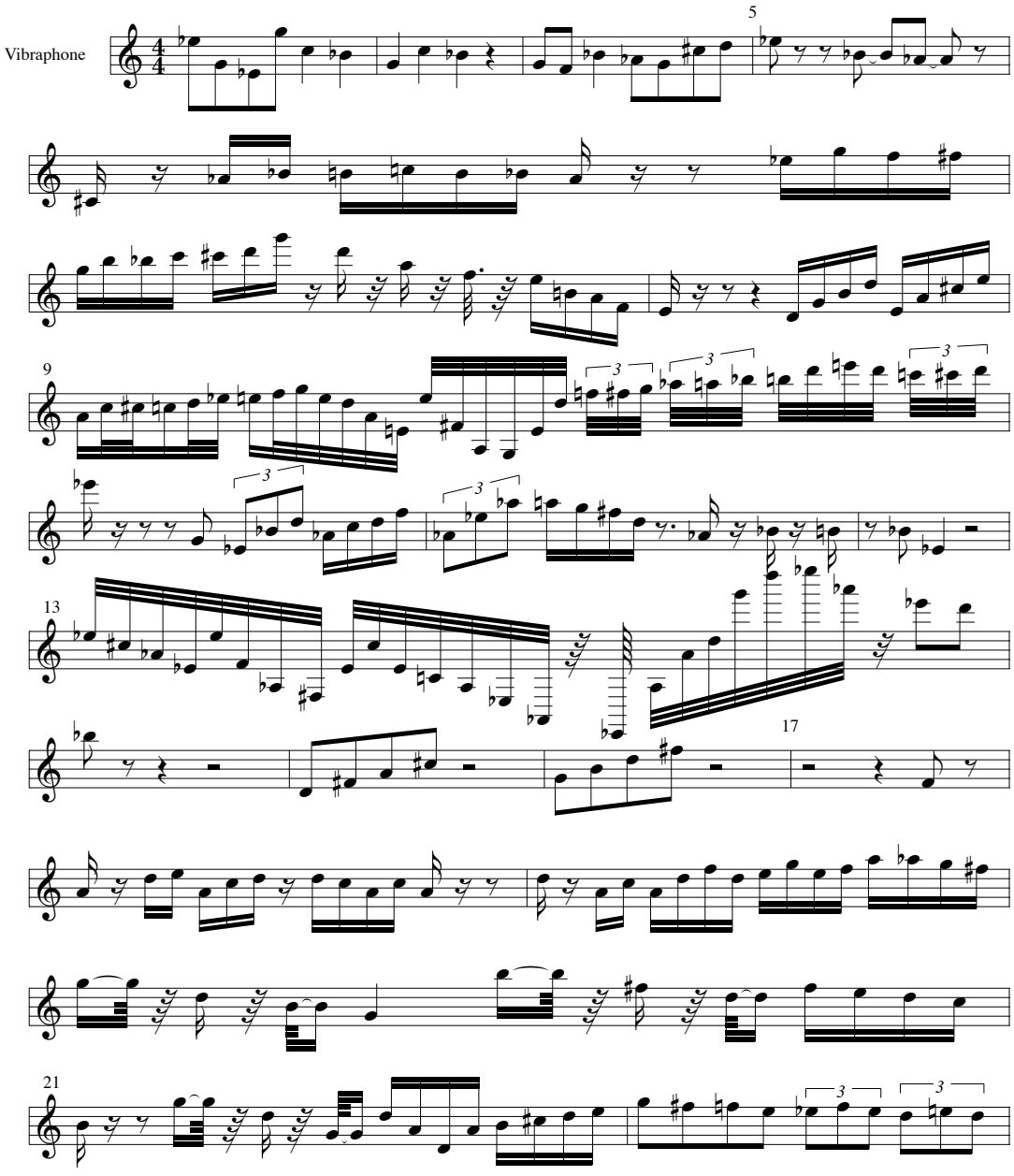


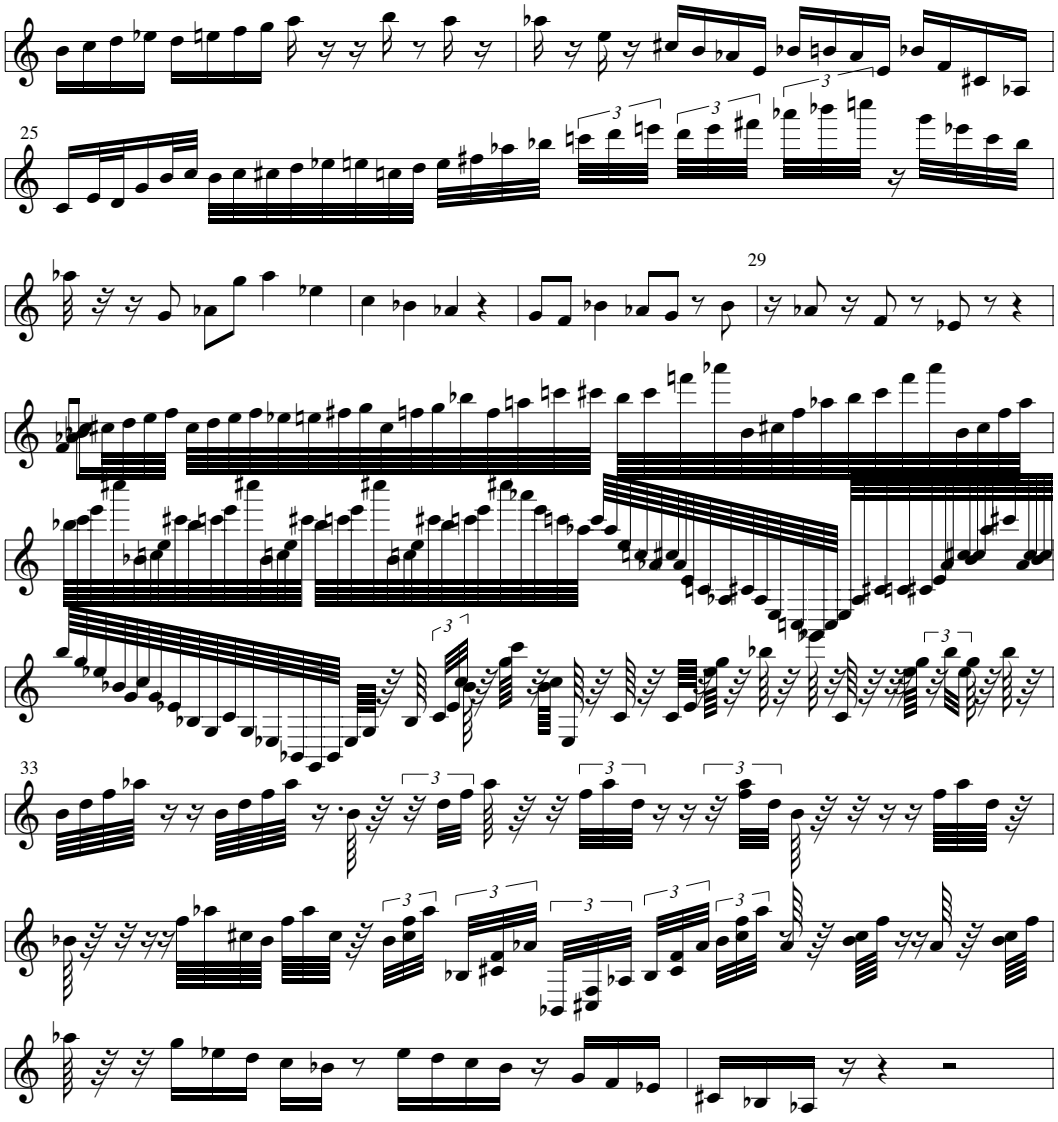

41
58
75
89
92
96
99
102
105
108


110

A.2 Hypothetical robot improvisation - All The Things You Are

Below is an improvisation generated by a hypothetical robot musician with a set of constraints allowing it to play monophonic lines extremely fast.


Vibraphone
5
9
13
17
21


25
29
33

APPENDIX B

SHIMON AND FRIENDS CONCERT SERIES

The *Shimon and Friends* concert is a show featuring Shimon, the Shimi robots, and robotic drumming prosthesis. Typically, a show consists of an hour long set with compositions by myself, Gil Weinberg, Govinda Ram-Pingali, Jason Barnes, and Chris Moore. Shows that extensively demonstrated the work of this thesis (particularly the path planning and improvisation system) are listed below:

1. West Lafayette, Indiana (February 18, 2016) - Purdue Convocations
2. Shanghai, China (October 28, 2016) - Shanghai International Interactive Arts Festival
3. Berlin, Germany (June 24, 2016) - Audi B͡eyond Summit
4. Durham, North Carolina (May 21, 2016) - Moogfest
5. Istanbul, Turkey (March 14, 2016) - Vodafone *Digital Transformations*
6. Atlanta, Georgia (September 11, 2015) - Georgia State *STEAM Cubed*
7. Brisbane, Australia (August 23, 2015) - Robotronica
8. Washington D.C., USA (July 22, 2015) - *25th Anniversary Celebration of the Americans with Disabilities Ac*t at The Kennedy Center


SHIMON ROBOT
& FRIENDS
MUSICAL ROBOTS AND CYBORGS FROM ROOM 100

SHIMON ROBOT & FRIENDS

Shimon is an improvising robotic musician that creates inspiring musical interactions with humans. Shimon listens to, understands, and collaborates with live human musicians in real time. Shimon uses artificial intelligence and creativity algorithms to push musical experiences to uncharted domains.

As an evening-length show, **SHIMON ROBOT & FRIENDS** features an ensemble comprised of two to five human musicians collaborating with Shimon robot (improvising on marimba) and Shimi robots that dance to sounds from smart phones. Together they perform original compositions ranging from jazz to hip hop, using artificial intelligence combined with human creativity, emotion, and aesthetic judgement.

Optional video excerpts during live performance, as well as talk-backs and educational workshops offered for all ages give the audience a fascinating behind-the-scenes look into how the robots were created and used in live performance.

Special guests including Jason Barnes, the cyborg drummer (see next page for further details), rappers and local musicians will be discussed and determined according to each presenter's needs and performance context.

THE CREATORS OF SHIMON ROBOT & FRIENDS

Shimon and Shimi robots were developed by the Georgia Institute of Technology and have been presented in dozens of concerts and festivals from DLD (Munich) to the US Science Festival, Google IO, TED, CNN, The Colbert Report, The Today Show, and most recently The Kennedy Center for the Performing Arts (Washington DC) and Robotronica (Australia.)

GIL WEINBERG is a musician and inventor of experimental musical instruments and robotic musicians. A professor of music technology at Georgia Tech and the founding director of the Georgia Tech Center for Music Technology, Weinberg's research focuses on artificial intelligence, human-robot interaction, assistive technology, and their application in music performance. His research led to over sixty scientific publications and four patents. Weinberg's music has been performed in festivals and concerts with orchestras such as the Deutsches Symphonie-Orchester Berlin, the National Irish Symphony Orchestra, and the Scottish BBC Symphony. His work has also been presented in venues such as the Cooper-Hewitt Museum, the Kennedy Center, Ars Electronica, the Boston Children's Museum, SIGGRAPH, DLD, and the World Economic Forum in Davos. Weinberg received his M.S. and Ph.D. degrees in Media Arts and Sciences from MIT and his Bachelor of Arts degree from the Interdisciplinary Program for Fostering Excellence in Tel Aviv University.

MASON BRETAN is a Ph.D. candidate of Music Technology and a member of Gil Weinberg's Robotic Musicianship Group. He is the primary developer and researcher for the Shimon, Shimi, and drumming prosthesis platforms. His research focuses on how to create robotic systems that are capable of meaningful musical interactions by designing sophisticated machine intelligence and robotic motion planning algorithms. Recently his work has been featured by Mashable, New Scientist, and The Washington Post. Bretan finished his undergraduate studies at UC San Diego in Interdisciplinary Computing and the Arts and is on schedule to finish his doctorate at Georgia Tech in Spring 2016. Mason performs guitar with SHIMON ROBOT & FRIENDS and a key role in the show's technical production from setup to execution.

SPECIAL GUEST: THE CYBORG DRUMMER

JASON BARNES was cleaning an exhaust duct when he was electrocuted by 22,000 volts of electricity and lost his arm. Seemingly, the musician's dreams of becoming a professional drummer were ended by the accident. When the drummer met Gil Weinberg from the Georgia Institute of Technology, the two men began to build a device that would allow Barnes to continue pursuing his dream at a high level. He is now the Cyborg Drummer who tours internationally as a guest on select SHIMON ROBOT & FRIENDS tour dates (subject to availability.)
Watch The Atlantic magazine's exclusive video interview.

APPENDIX C

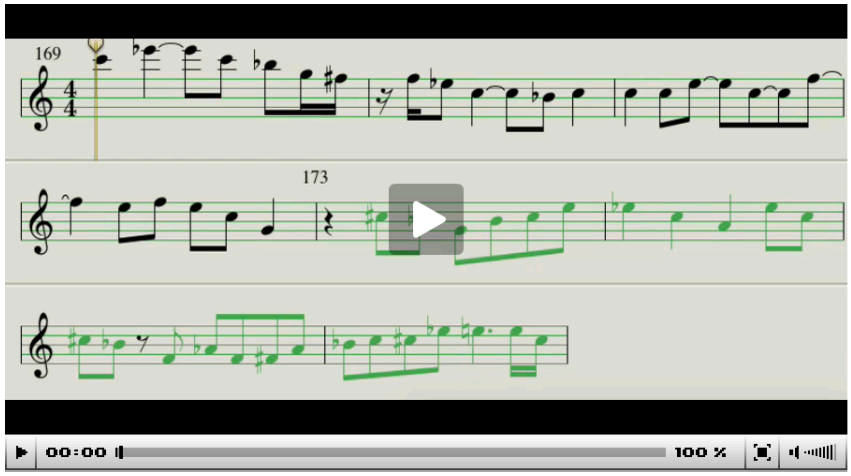
SURVEY

A sample from one-page of the survey used in the subjective listening test. The survey was developed using Qualtrics Survey Software (compliments of Georgia Tech subscription).

Music Generation Forced Choice

You are being asked to evaluate a computer music generation system. On each page you will listen to four different musical clips and then rank the clips according to the criteria asked of you. To rank the clips simply drag the item in the list to where you would like it to be. You will listen to and rank 10 different sets of musical clips.

Clip 1



Clip 2

Clip 3

Clip 4

Q2 **Naturalness**.Rank the four clips according to the naturalness of the transition between the first four measures (in black) and the second four measures (in green). Rank in decreasing order so that the most natural is at the top.

_____ Clip 1 (1)
_____ Clip 2 (2)
_____ Clip 3 (3)
_____ Clip 4 (4)

Q3 **Naturalness**.Rank the four clips according to how likely or appropriate that the musical content of the last four measures (in green) follows the first four

measures (in black). Rank in decreasing order so that the most appropriate is at the top.

_____ Clip 1 (1)
_____ Clip 2 (2)
_____ Clip 3 (3)
_____ Clip 4 (4)

Q4 **Naturalness**.Considering only the last four measures (in green) of each clip, rank the naturalness of these measures in term of the overall rhythmic structure, overall pitch contours, and appropriateness of individual note transitions. Rank in decreasing order so that the most natural is at the top.

_____ Clip 1 (1)
_____ Clip 2 (2)
_____ Clip 3 (3)
_____ Clip 4 (4)

Q5 **Likeability**.Considering only the last four measures (in green) of each clip, rank according to how much you liked the musical content of these measures. Rank in decreasing order so that the most liked is at the top.

_____ Clip 1 (1)
_____ Clip 2 (2)
_____ Clip 3 (3)
_____ Clip 4 (4)

Q6 **Likeability**.Consider all eight measures of each clip, rank according to how much you like the overall content of the entire clip. Rank in decreasing order so that the most liked is at the top.

_____ Clip 1 (1)
_____ Clip 2 (2)
_____ Clip 3 (3)
_____ Clip 4 (4)

REFERENCES

[1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., and others, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[2] Adler, S. and Hesterman, P., *The study of orchestration*. WW Norton, 1989.

[3] Albin, A., Weinberg, G., and Egerstedt, M., "Musical abstractions in distributed multi-robot systems," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 451–458, IEEE, 2012.

[4] Almeida, A., George, D., Smith, J., and Wolfe, J., "The clarinet: How blowing pressure, lip force, lip position and reed ?hardness? affect pitch, sound level, and spectrum," *The Journal of the Acoustical Society of America*, vol. 134, no. 3, pp. 2247–2255, 2013.

[5] Anderson, M. L., "Embodied cognition: A field guide," *Artificial intelligence*, vol. 149, no. 1, pp. 91–130, 2003.

[6] Asada, M., Hosoda, K., Kuniyoshi, Y., Ishiguro, H., Inui, T., Yoshikawa, Y., Ogino, M., and Yoshida, C., "Cognitive developmental robotics: a survey," *Autonomous Mental Development, IEEE Transactions on*, vol. 1, no. 1, pp. 12–34, 2009.

[7] Atkeson, C. G., Hale, J. G., Pollick, F. E., Riley, M., Kotosaka, S., Schaul, S., Shibata, T., Tevatia, G., Ude, A., Vijayakumar, S., and others, "Using humanoid robots to study human behavior," *IEEE Intelligent Systems and their applications*, vol. 15, no. 4, pp. 46–56, 2000.

[8] Baginsky, N., "The three sirens: A self learning robotic rock band," *Available online at http://www.the-three-sirens.info/binfo.html*, accessed Oct 2, 2014.

[9] Balcetis, E. and Dunning, D., "Cognitive dissonance and the perception of natural environments," *Psychological Science*, vol. 18, no. 10, pp. 917–921, 2007.

[10] Barraquand, J. and Latombe, J.-C., "A monte-carlo algorithm for path planning with many degrees of freedom," in *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, pp. 1712–1717, IEEE, 1990.

[11] Barton, S., "The human, the mechanical, and the spaces in between: Explorations in human-robotic musical improvisation," in *Ninth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2013.

[12] Batula, A. M. and Kim, Y. E., "Development of a mini-humanoid pianist," in *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, pp. 192–197, IEEE, 2010.

[13] Baumann, S., Koeneke, S., Schmidt, C. F., Meyer, M., Lutz, K., and Jancke, L., "A network for audio–motor coordination in skilled pianists and non-musicians," *Brain research*, vol. 1161, pp. 65–78, 2007.

[14] Bekkering, H. and Neggers, S. F., "Visual search is modulated by action intentions," *Psychological science*, vol. 13, no. 4, pp. 370–374, 2002.

[15] Bello, J. P., Daudet, L., Abdallah, S., Duxbury, C., Davies, M., and Sandler, M. B., "A tutorial on onset detection in music signals," *Speech and Audio Processing, IEEE Transactions on*, vol. 13, no. 5, pp. 1035–1047, 2005.

[16] Benetos, E. and Holzapfel, A., "Automatic transcription of turkish microtonal music," *The Journal of the Acoustical Society of America*, vol. 138, no. 4, pp. 2118–2130, 2015.

[17] Bicchi, A., Gabiccini, M., and Santello, M., "Modelling natural and artificial hands with synergies," *Phil. Trans. R. Soc. B*, vol. 366, no. 1581, pp. 3153–3161, 2011.

[18] Biles, J., "Genjam: A genetic algorithm for generating jazz solos," in *Proceedings of the International Computer Music Conference*, pp. 131–131, INTERNATIONAL COMPUTER MUSIC ACCOCIATION, 1994.

[19] Black, A. W. and Taylor, P. A., "Automatically clustering similar units for unit selection in speech synthesis.," 1997.

[20] Boulanger-Lewandowski, N., Bengio, Y., and Vincent, P., "Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription," *arXiv preprint arXiv:1206.6392*, 2012.

[21] Bower, G. H., "Mental imagery and associative learning," *Cognition in learning and memory. New York: Wiley*, vol. 1372, pp. 51–88, 1972.

[22] Bown, O., Eldridge, A., and McCormack, J., "Understanding interaction in contemporary digital music: from instruments to behavioural objects," *Organised Sound*, vol. 14, no. 02, pp. 188–196, 2009.

[23] Breazeal, C., Gray, J., and Berlin, M., "An embodied cognition approach to mindreading skills for socially intelligent robots," *The International Journal of Robotics Research*, vol. 28, no. 5, pp. 656–680, 2009.

[24] Bretan, M., Cicconet, M., Nikolaidis, R., and Weinberg, G., "Developing and composing for a robotic musician," in *Proc. International Computer Music Conference on (ICMC'12)*, (Ljubljana, Slovenia), Sept. 2012.

[25] Bretan, M., Gopinath, D., Mullins, P., and Weinberg, G., "A robotic prosthesis for an amputee drummer," *arXiv preprint arXiv:1612.04391*, 2016.

[26] Bretan, M., Gopinath, D., Mullins, P., and Weinberg, G., "A robotic prosthesis for an amputee drummer," *Journal of Human-Robotic Interaction*, in review.

[27] Bretan, M., Hoffman, G., and Weinberg, G., "Emotionally expressive dynamic physical behaviors in robots," *International Journal of Human-Computer Studies*, 2016.

[28] Bretan, M. and Weinberg, G., "Chronicles of a robotic musical companion," in *Proceedings of the 2014 conference on New interfaces for musical expression, University of London*, 2014.

[29] Bretan, M. and Weinberg, G., "A survey of robotic musicianship," *Communications of the ACM*, vol. 59, no. 5, pp. 100–109, 2016.

[30] Brooks, R. A., "Intelligence without representation," *Artificial intelligence*, vol. 47, no. 1-3, pp. 139–159, 1991.

[31] Chadefaux, D., Le Carrou, J.-L., Vitrani, M.-A., Billout, S., and Quartier, L., "Harp plucking robotic finger," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 4886–4891, IEEE, 2012.

[32] Chapados, C. and Levitin, D. J., "Cross-modal interactions in the experience of musical performances: Physiological correlates," *Cognition*, vol. 108, no. 3, pp. 639–651, 2008.

[33] Chapel, R. H., "Realtime algorithmic music systems from fractals and chaotic functions: Towards an active musical instrument," *Barcelona: Universitat Pompeu Fabra*, 2003.

[34] Chen, R., Shen, W., Srinivasamurthy, A., and Chordia, P., "Chord recognition using duration-explicit hidden markov models.," in *ISMIR*, pp. 445–450, 2012.

[35] Cheng, G., Hyon, S.-H., Morimoto, J., Ude, A., Hale, J. G., Colvin, G., Scroggin, W., and Jacobsen, S. C., "Cb: A humanoid research platform for exploring neuroscience," *Advanced Robotics*, vol. 21, no. 10, pp. 1097–1114, 2007.

[36] Ching, W.-K., Ng, M. K., and Fung, E. S., "Higher-order multivariate markov chains and their applications," *Linear Algebra and its Applications*, vol. 428, no. 2-3, pp. 492–507, 2008.

[37] Chordia, P., Sastry, A., and Şentürk, S., "Predictive tabla modelling using variable-length markov and hidden markov models," *Journal of New Music Research*, vol. 40, no. 2, pp. 105–118, 2011.

[38] Cicconet, M., Bretan, M., and Weinberg, G., "Human-robot percussion ensemble: Anticipation on the basis of visual cues," *Robotics & Automation Magazine, IEEE*, vol. 20, no. 4, pp. 105–110, 2013.

[39] Clayton, M., Sager, R., and Will, U., "In time with the music: The concept of entrainment and its significance for ethnomusicology," in *European meetings in ethnomusicology*, vol. 11, pp. 3–142, 2005.

[40] Coca, A. E., Romero, R. A., and Zhao, L., "Generation of composed musical structures through recurrent neural networks based on chaotic inspiration," in *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pp. 3220–3226, IEEE, 2011.

[41] Collins, N. M., *Towards autonomous agents for live computer music: Realtime machine listening and interactive music systems*. PhD thesis, Citeseer, 2006.

[42] Collins, T., Laney, R., Willis, A., and Garthwaite, P. H., "Developing and evaluating computational models of musical style," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 30, no. 01, pp. 16–43, 2016.

[43] Conkie, A., Beutnagel, M. C., Syrdal, A. K., and Brown, P. E., "Preselection of candidate units in a unit selection-based text-to-speech synthesis system," in *Proc. ICSLP, Beijing*, 2000.

[44] Cope, D., "Techniques of the contemporary composer," 1997.

[45] Cope, D., "One approach to musical intelligence," *IEEE Intelligent systems and their applications*, vol. 14, no. 3, pp. 21–25, 1999.

[46] Cope, D. and Mayer, M. J., *Experiments in musical intelligence*, vol. 12. AR editions Madison, WI, 1996.

[47] Crook, H., *How to improvise: an approach to practising improvisation*. Advance music, 1991.

[48] Dahl, S. and Friberg, A., "Visual perception of expressiveness in musicians' body movements," 2007.

[49] Dannenberg, R. B., Brown, B., Zeglin, G., and Lupish, R., "Mcblare: a robotic bagpipe player," in *Proceedings of the 2005 conference on New interfaces for musical expression*, pp. 80–84, National University of Singapore, 2005.

[50] Dannenberg, R. B., Brown, H. B., and Lupish, R., "Mcblare: A robotic bagpipe player," in *Musical Robots and Interactive Multimodal Systems*, pp. 165–178, Springer, 2011.

[51] Davies, M. E. and Plumbley, M. D., "Causal tempo tracking of audio.," in *ISMIR*, 2004.

[52] Des, M., "Django?s hand," *BMJ*, vol. 339, p. 1427, 2009.

[53] Dregni, M., Antonietto, A., and Legrand, A., *Django Reinhardt and the illustrated history of gypsy jazz*. Fulcrum Publishing, 2006.

[54] Drost, U. C., Rieger, M., Brass, M., Gunter, T. C., and Prinz, W., "When hearing turns into playing: Movement induction by auditory stimuli in pianists," *The Quarterly Journal of Experimental Psychology Section A*, vol. 58, no. 8, pp. 1376–1389, 2005.

[55] Drozdov, I., Kidd, M., and Modlin, I. M., "Evolution of one-handed piano compositions," *The Journal of hand surgery*, vol. 33, no. 5, pp. 780–786, 2008.

[56] Drummond, J., "Understanding interactive systems," *Organised Sound*, vol. 14, no. 02, pp. 124–133, 2009.

[57] Dubnov, S., Assayag, G., Lartillot, O., and Bejerano, G., "Using machine-learning methods for musical style modeling," *Computer*, vol. 36, no. 10, pp. 73–80, 2003.

[58] Eck, D. and Schmidhuber, J., "Finding temporal structure in music: Blues improvisation with lstm recurrent networks," in *Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on*, pp. 747–756, IEEE, 2002.

[59] Edakkattil Gopinath, D., *Enhancing stroke generation and expressivity in robotic drummers-A generative physics model approach*. PhD thesis, Georgia Institute of Technology, 2015.

[60] Eitan, Z., *Highpoints: A study of melodic peaks*. University of Pennsylvania Press Philadelphia, PA, 1997.

[61] Ferrand, D. and Vergez, C., "Blowing machine for wind musical instrument: toward a real-time control of the blowing pressure," in *Control and Automation, 2008 16th Mediterranean Conference on*, pp. 1562–1567, IEEE, 2008.

[62] Fidlon, J. D., *Cognitive dimensions of instrumental jazz improvisation*. PhD thesis, 2011.

[63] Foglia, L. and Wilson, R. A., "Embodied cognition," *Wiley Interdisciplinary Reviews: Cognitive Science*, vol. 4, no. 3, pp. 319–325, 2013.

[64] Franklin, J. A., "Multi-phase learning for jazz improvisation and interaction," in *Proceedings of the Eighth Biennial Symposium for Arts & Technology*, 2001.

[65] Franklin, J. A., "Recurrent neural networks for music computation," *INFORMS Journal on Computing*, vol. 18, no. 3, pp. 321–338, 2006.

[66] Gabiccini, M., Bicchi, A., Prattichizzo, D., and Malvezzi, M., "On the role of hand synergies in the optimal choice of grasping forces," *Autonomous Robots*, vol. 31, no. 2-3, p. 235, 2011.

[67] Gallese, V., Keysers, C., and Rizzolatti, G., "A unifying view of the basis of social cognition," *Trends in cognitive sciences*, vol. 8, no. 9, pp. 396–403, 2004.

[68] Gibson, W., "Material culture and embodied action: sociological notes on the examination of musical instruments in jazz improvisation," *The Sociological Review*, vol. 54, no. 1, pp. 171–187, 2006.

[69] Godøy, R. I. and Leman, M., *Musical gestures: Sound, movement, and meaning*. Routledge, 2010.

[70] Goel, K., Vohra, R., and Sahoo, J., "Polyphonic music generation by modeling temporal dependencies using a rnn-dbn," in *Artificial Neural Networks and Machine Learning–ICANN 2014*, pp. 217–224, Springer, 2014.

[71] Goldin-Meadow, S., Nusbaum, H., Kelly, S. D., and Wagner, S., "Explaining math: Gesturing lightens the load," *Psychological Science*, vol. 12, no. 6, pp. 516–522, 2001.

[72] Goldman, A., "What does one know when one knows how to improvise,"

[73] Goldman, A., "Towards a cognitive–scientific research program for improvisation: Theory and an experiment.," *Psychomusicology: Music, Mind, and Brain*, vol. 23, no. 4, p. 210, 2013.

[74] Gray, W. D. and Veksler, V. D., "The acquisition and asymmetric transfer of interactive routines," in *Proceedings of the Cognitive Science Society*, vol. 27, 2005.

[75] Guthrie, D., Allison, B., Liu, W., Guthrie, L., and Wilks, Y., "A closer look at skip-gram modelling," in *Proceedings of the 5th international Conference on Language Resources and Evaluation (LREC-2006)*, pp. 1–4, 2006.

[76] Harker, B., "" telling a story": Louis armstrong and coherence in early jazz," *Current Musicology*, vol. 63, p. 46, 1997.

[77] Harwood, D. L., "Universals in music: A perspective from cognitive psychology," *Ethnomusicology*, pp. 521–533, 1976.

[78] Haslinger, B., Erhard, P., Altenmüller, E., Schroeder, U., Boecker, H., and Ceballos-Baumann, A. O., "Transmodal sensorimotor networks during action observation in professional pianists," *Journal of cognitive neuroscience*, vol. 17, no. 2, pp. 282–293, 2005.

[79] Hewlett, W. B. and Selfridge-Field, E., *Melodic similarity: Concepts, procedures, and applications*, vol. 11. The MIT Press, 1998.

[80] Hochenbaum, J. and Kapur, A., "Drum stroke computing: Multimodal signal processing for drum stroke identification and performance metrics," in *International Conference on New Interfaces for Musical Expression*, 2012.

[81] Hoffman, G., "Embodied cognition for autonomous interactive robots," *Topics in cognitive science*, vol. 4, no. 4, pp. 759–772, 2012.

[82] Hoffman, G. and Weinberg, G., "Synchronization in human-robot musicianship," in *RO-MAN, 2010 IEEE*, pp. 718–724, IEEE, 2010.

[83] Hoffman, G. and Weinberg, G., "Interactive improvisation with a robotic marimba player," *Autonomous Robots*, vol. 31, no. 2-3, pp. 133–153, 2011.

[84] Howe, B., "Paul wittgenstein and the performance of disability," *The Journal of Musicology*, vol. 27, no. 2, pp. 135–180, 2010.

[85] Huang, C.-Z. A., Duvenaud, D., and Gajos, K. Z., "Chordripple: Recommending chords to help novice composers go beyond the ordinary," in *Proceedings of the 21st International Conference on Intelligent User Interfaces*, pp. 241–250, ACM, 2016.

[86] Huang, P.-S., He, X., Gao, J., Deng, L., Acero, A., and Heck, L., "Learning deep structured semantic models for web search using clickthrough data," in *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pp. 2333–2338, ACM, 2013.

[87] Humphrey, E. J., Cho, T., and Bello, J. P., "Learning a robust tonnetz-space transform for automatic chord recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pp. 453–456, IEEE, 2012.

[88] Hunt, A. J. and Black, A. W., "Unit selection in a concatenative speech synthesis system using a large speech database," in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, vol. 1, pp. 373–376, IEEE, 1996.

[89] Hurley, S., "Perception and action: Alternative views," *Synthese*, vol. 129, no. 1, pp. 3–40, 2001.

[90] Iordanescu, L., Grabowecky, M., and Suzuki, S., "Action enhances auditory but not visual temporal sensitivity," *Psychonomic bulletin & review*, vol. 20, no. 1, pp. 108–114, 2013.

[91] Iyer, V., "Embodied mind, situated cognition, and expressive microtiming in african-american music," *Music Perception: An Interdisciplinary Journal*, vol. 19, no. 3, pp. 387–414, 2002.

[92] Jehan, T., *Creating music by listening*. PhD thesis, Massachusetts Institute of Technology, 2005.

[93] Jensen, M. G., "John cage, chance operations, and the chaos game: Cage and the" i ching"," *The Musical Times*, vol. 150, no. 1907, pp. 97–102, 2009.

[94] Jensenius, A. R., "An action–sound approach to teaching interactive music," *Organised Sound*, vol. 18, no. 02, pp. 178–189, 2013.

[95] Jo, W., Lee, B., and Kim, D., "Development of auditory feedback system for violin playing robot," *International Journal of Precision Engineering and Manufacturing*, vol. 17, no. 6, pp. 717–724, 2016.

[96] Johnson-Laird, P. N., "How jazz musicians improvise," *Music Perception: An Interdisciplinary Journal*, vol. 19, no. 3, pp. 415–442, 2002.

[97] Jordà, S., "Afasia: the ultimate homeric one-man-multimedia-band," in *Proceedings of the 2002 conference on New interfaces for musical expression*, pp. 1–6, National University of Singapore, 2002.

[98] Kahn Jr, P. H., Kanda, T., Ishiguro, H., Freier, N. G., Severson, R. L., Gill, B. T., Ruckert, J. H., and Shen, S., "robovie, you'll have to go into the closet now: Children's social and moral relationships with a humanoid robot.," *Developmental psychology*, vol. 48, no. 2, p. 303, 2012.

[99] Kapoor, A. and Taylor, R. H., "A constrained optimization approach to virtual fixtures for multi-handed tasks," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pp. 3401–3406, IEEE, 2008.

[100] Kapur, A., "A history of robotic musical instruments," in *Proceedings of the International Computer Music Conference*, pp. 21–28, Citeseer, 2005.

[101] Kapur, A., "Multimodal techniques for human/robot interaction," in *Musical Robots and Interactive Multimodal Systems*, pp. 215–232, Springer, 2011.

[102] Kapur, A., Darling, M., Diakopoulos, D., Murphy, J. W., Hochenbaum, J., Vallis, O., and Bahn, C., "The machine orchestra: An ensemble of human laptop performers and robotic musical instruments," *Computer Music Journal*, vol. 35, no. 4, pp. 49–63, 2011.

[103] Kapur, A., Lazier, A. J., Davidson, P., Wilson, R. S., and Cook, P. R., "The electronic sitar controller," in *Proceedings of the 2004 conference on New interfaces for musical expression*, pp. 7–12, National University of Singapore, 2004.

[104] Kapur, A., Murphy, J., and Carnegie, D., "Kritaanjli: A robotic harmonium for performance, pedogogy and research,"

[105] Kapur, A., Trimpin, E. S., Suleman, A., and Tzanetakis, G., "A comparison of solenoid-based strategies for robotic drumming," *ICMC, Copenhagen, Denmark*, 2007.

[106] Katayose, H., Hashida, M., De Poli, G., and Hirata, K., "On evaluating systems for generating expressive music performance: the rencon experience," *Journal of New Music Research*, vol. 41, no. 4, pp. 299–310, 2012.

[107] Kato, I., Ohteru, S., Shirai, K., Matsushima, T., Narita, S., Sugano, S., Kobayashi, T., and Fujisawa, E., "The robot musician 'wabot-2'(waseda robot-2)," *Robotics*, vol. 3, no. 2, pp. 143–155, 1987.

[108] Keller, R. M. and Morrison, D. R., "A grammatical approach to automatic improvisation," in *Proceedings, Fourth Sound and Music Conference, Lefkada, Greece, July.Most of the soloists at Birdland had to wait for Parkers next record in order to find out what to play next. What will they do now*, 2007.

[109] Khatib, O., Yokoi, K., Chang, K., Ruspini, D., Holmberg, R., and Casal, A., "Coordination and decentralized cooperation of multiple mobile manipulators," *Journal of Field Robotics*, vol. 13, no. 11, pp. 755–764, 1996.

[110] Kidd, C. and Breazeal, C., "Comparison of social presence in robots and animated characters," *Proc of human-computer interaction (CHI)*, 2005.

[111] Kidd, C. D. and Breazeal, C., "Effect of a robot on user perceptions," in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 4, pp. 3559–3564, IEEE, 2004.

[112] Kilner, J. M., Friston, K. J., and Frith, C. D., "Predictive coding: an account of the mirror neuron system," *Cognitive processing*, vol. 8, no. 3, pp. 159–166, 2007.

[113] Kim, Y. E., Batula, A. M., Grunberg, D., Lofaro, D. M., Oh, J., and Oh, P. Y., "Developing humanoids for musical interaction," in *International Conference on Intelligent Robots and Systems*, 2010.

[114] Kim, Y. E., Schmidt, E. M., Migneco, R., Morton, B. G., Richardson, P., Scott, J., Speck, J. A., and Turnbull, D., "Music emotion recognition: A state of the art review," in *Proc. ISMIR*, pp. 255–266, Citeseer, 2010.

[115] Kirke, A. and Miranda, E. R., "A survey of computer systems for expressive music performance," *ACM Computing Surveys (CSUR)*, vol. 42, no. 1, p. 3, 2009.

[116] Klapuri, A., "Sound onset detection by applying psychoacoustic knowledge," in *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, vol. 6, pp. 3089–3092, IEEE, 1999.

[117] Kuniyoshi, Y., Yorozu, Y., Suzuki, S., Sangawa, S., Ohmura, Y., Terada, K., and Nagakubo, A., "Emergence and development of embodied cognition: A constructivist approach using robots," *Progress in brain research*, vol. 164, pp. 425–445, 2007.

[118] Latombe, J.-C., *Robot motion planning*, vol. 124. Springer Science & Business Media, 2012.

[119] Leach, J. and Fitch, J., "Nature, music, and algorithmic composition," *Computer Music Journal*, vol. 19, no. 2, pp. 23–33, 1995.

[120] Lee, S.-l., Lau, I. Y.-m., Kiesler, S., and Chiu, C.-Y., "Human mental models of humanoid robots," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pp. 2767–2772, IEEE, 2005.

[121] Leman, M., *Embodied music cognition and mediation technology*. Mit Press, 2008.

[122] Leman, M., Desmet, F., Styns, F., Van Noorden, L., and Moelants, D., "Sharing musical expression through embodied listening: a case study based on chinese guqin music," 2009.

[123] Leong, T. W., Vetere, F., and Howard, S., "Randomness as a resource for design," in *Proceedings of the 6th conference on Designing Interactive systems*, pp. 132–139, ACM, 2006.

[124] Lerdahl, F., "Cognitive constraints on compositional systems," *Contemporary Music Review*, vol. 6, no. 2, pp. 97–121, 1992.

[125] Lerdahl, F. and Jackendoff, R., *A generative theory of tonal music*. MIT press, 1985.

[126] Lerdahl, F. and Jackendoff, R., "A generative theory of tonal music," 1987.

[127] Lerdahl, F. and Krumhansl, C. L., "Modeling tonal tension," 2007.

[128] Levine, M., *The jazz piano book*. " O'Reilly Media, Inc.", 2011.

[129] Levy, O. and Goldberg, Y., "Dependency-based word embeddings.," in *ACL (2)*, pp. 302–308, Citeseer, 2014.

[130] Lewis, G. E., "Too many notes: Computers, complexity and culture in voyager," *Leonardo Music Journal*, vol. 10, pp. 33–39, 2000.

[131] Lim, A., Mizumoto, T., Cahier, L.-K., Otsuka, T., Takahashi, T., Komatani, K., Ogata, T., and Okuno, H. G., "Robot musical accompaniment: integrating audio and visual cues for real-time synchronization with a human flutist," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 1964–1969, IEEE, 2010.

[132] Livingstone, S. R., Thompson, W. F., and Russo, F. A., "Facial expressions and emotional singing: A study of perception and production with motion capture and electromyography," 2009.

[133] Logan-Greene, R., "The music of richard johnson logan-greene." `http://zownts.com`. Accessed: 5-1-2014.

[134] Logan-Greene, R., *Submersions I*. University of Washington, 2011.

[135] Maciejewski, A. A. and Klein, C. A., "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *The international journal of robotics research*, vol. 4, no. 3, pp. 109–117, 1985.

[136] Maes, L., Raes, G.-W., and Rogers, T., "The man and machine robot orchestra at logos," *Computer Music Journal*, vol. 35, no. 4, pp. 28–48, 2011.

[137] Maes, P.-J., Leman, M., Palmer, C., and Wanderley, M., "Action-based effects on music perception," *Frontiers in psychology*, vol. 4, p. 1008, 2014.

[138] Manning, F. and Schutz, M., "?moving to the beat? improves timing perception," *Psychonomic bulletin & review*, vol. 20, no. 6, pp. 1133–1139, 2013.

[139] McClelland, J. L., Botvinick, M. M., Noelle, D. C., Plaut, D. C., Rogers, T. T., Seidenberg, M. S., and Smith, L. B., "Letting structure emerge: connectionist and dynamical systems approaches to cognition," *Trends in cognitive sciences*, vol. 14, no. 8, pp. 348–356, 2010.

[140] McCormack, J., "Grammar based music composition," *Complex systems*, vol. 96, pp. 321–336, 1996.

[141] McPherson, A., "The magnetic resonator piano: Electronic augmentation of an acoustic grand piano," *Journal of New Music Research*, vol. 39, no. 3, pp. 189–202, 2010.

[142] McVay, J., Carnegie, D., Murphy, J., and Kapur, A., "Mechbass: A systems overview of a new four-stringed robotic bass guitar," in *Proceedings of the 2012 Electronics New Zealand Conference, Dunedin, New Zealand*, 2012.

[143] Menzies, D. W. and McPherson, A., "An electronic bagpipe chanter for automatic recognition of highland piping ornamentation," *Proc. NIME, Ann Arbor, MI, USA*, 2012.

[144] Metheny, P., "Orchestrion." http://www.theorchestrionproject.com/. Accessed: 5-1-2014.

[145] Metta, G., Sandini, G., Vernon, D., Natale, L., and Nori, F., "The icub humanoid robot: an open platform for research in embodied cognition," in *Proceedings of the 8th workshop on performance metrics for intelligent systems*, pp. 50–56, ACM, 2008.

[146] Miranda, E. R. and Tikhanoff, V., "Musical composition by autonomous robots: A case study with aibo," *Proceedings of TAROS 2005 (Towards Autonomous Robotic Systems)*, 2005.

[147] Mizumoto, T., Lim, A., Otsuka, T., Nakadai, K., Takahashi, T., Ogata, T., and Okuno, H. G., "Integration of flutist gesture recognition and beat tracking for human-robot ensemble," in *Proc of IEEE/RSJ-2010 Workshop on Robots and Musical Expression*, pp. 159–171, 2010.

[148] Nelson, A. L., Barlow, G. J., and Doitsidis, L., "Fitness functions in evolutionary robotics: A survey and analysis," *Robotics and Autonomous Systems*, vol. 57, no. 4, pp. 345–370, 2009.

[149] Newell, A., Shaw, J., and Simon, H. A., "Chess-playing programs and the problem of complexity," in *Computer Games I*, pp. 89–115, Springer, 1988.

[150] Nikolaidis, R. and Weinberg, G., "Playing with the masters: A model for improvisatory musical interaction between robots and humans," in *RO-MAN, 2010 IEEE*, pp. 712–717, IEEE, 2010.

[151] Norgaard, M., "Descriptions of improvisational thinking by artist-level jazz musicians," *Journal of Research in Music Education*, vol. 59, no. 2, pp. 109–127, 2011.

[152] Norgaard, M., "How jazz musicians improvise," *Music Perception: An Interdisciplinary Journal*, vol. 31, no. 3, pp. 271–287, 2014.

[153] Norgaard, M., "Descriptions of improvisational thinking by developing jazz improvisers," *International Journal of Music Education*, p. 0255761416659512, 2016.

[154] Nusseck, M. and Wanderley, M. M., "Music and motionhow music-related ancillary body movements contribute to the experience of music," 2009.

[155] Otsuka, T., Nakadai, K., Takahashi, T., Ogata, T., and Okuno, H. G., "Real-time audio-to-score alignment using particle filter for coplayer music robots," *EURASIP Journal on Advances in Signal Processing*, vol. 2011, p. 2, 2011.

[156] Pachet, F. and Roy, P., "Markov constraints: steerable generation of markov sequences," *Constraints*, vol. 16, no. 2, pp. 148–172, 2011.

[157] Paiva, R. P., Mendes, T., and Cardoso, A., "On the detection of melody notes in polyphonic audio.," in *ISMIR*, pp. 175–182, 2005.

[158] Pan, Y., Kim, M.-G., and Suzuki, K., "A robot musician interacting with a human partner through initiative exchange," in *Proc of Conf on New Interfaces for Musical Expression*, pp. 166–169, 2010.

[159] Papadopoulos, G. and Wiggins, G., "Ai methods for algorithmic composition: A survey, a critical view and future prospects," in *AISB Symposium on Musical Creativity*, pp. 110–117, Edinburgh, UK, 1999.

[160] Phillips-Silver, J. and Trainor, L. J., "Feeling the beat: movement influences infant rhythm perception," *Science*, vol. 308, no. 5727, pp. 1430–1430, 2005.

[161] Phillips-Silver, J. and Trainor, L. J., "Hearing what the body feels: Auditory encoding of rhythmic movement," *Cognition*, vol. 105, no. 3, pp. 533–546, 2007.

[162] Piazza, C., Della Santina, C., Catalano, M., Grioli, G., Garabini, M., and Bicchi, A., "Softhand pro-d: Matching dynamic content of natural user commands with hand embodiment for enhanced prosthesis control," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pp. 3516–3523, IEEE, 2016.

[163] Pressing, J., "Improvisation: methods and models," *John A. Sloboda (Hg.): Generative processes in music, Oxford*, pp. 129–178, 1988.

[164] Raes, G., "Robots and automatons catalogue," *Available online at http://logosfoundation.org/instrumg_wr/automatons.html*, accessed Oct 2, 2014.

[165] Repp, B. H. and Knoblich, G., "Performed or observed keyboard actions affect pianists' judgements of relative pitch," *The Quarterly Journal of Experimental Psychology*, vol. 62, no. 11, pp. 2156–2170, 2009.

[166] Rizzolatti, G., "The mirror neuron system and its function in humans," *Anatomy and embryology*, vol. 210, no. 5-6, pp. 419–421, 2005.

[167] Rizzolatti, G. and Craighero, L., "The mirror-neuron system," *Annu. Rev. Neurosci.*, vol. 27, pp. 169–192, 2004.

[168] Rogers, T., Kemper, S., and Barton, S., "Emmi: Expressive machines musical instruments," *Available online at http://expressivemachines.com/ (accessed Oct 6, 2014)*, 2014.

[169] Rowe, R., *Machine musicianship*. MIT press, 2004.

[170] Sariff, N. and Buniyamin, N., "An overview of autonomous mobile robot path planning algorithms," in *Research and Development, 2006. SCOReD 2006. 4th Student Conference on*, pp. 183–188, IEEE, 2006.

[171] Sawyer, K., "Improvisational creativity: An analysis of jazz performance," *Creativity Research Journal*, vol. 5, no. 3, pp. 253–263, 1992.

[172] Schuller, G., *Early jazz: Its roots and musical development*, vol. 1. Oxford University Press, USA, 1986.

[173] Sedlmeier, P., Weigelt, O., and Walther, E., "Music is in the muscle: How embodied cognition may influence music preferences," *Music Perception: An Interdisciplinary Journal*, vol. 28, no. 3, pp. 297–306, 2011.

[174] Shadmehr, R. and Mussa-Ivaldi, F. A., "Adaptive representation of dynamics during learning of a motor task," *The Journal of Neuroscience*, vol. 14, no. 5, pp. 3208–3224, 1994.

[175] Sheh, A. and Ellis, D. P., "Chord segmentation and recognition using em-trained hidden markov models," *ISMIR 2003*, pp. 185–191, 2003.

[176] Shibuya, K., Ideguchi, H., and Ikushima, K., "Volume control by adjusting wrist moment of violin-playing robot," *International Journal of Synthetic Emotions (IJSE)*, vol. 3, no. 2, pp. 31–47, 2012.

[177] Shibuya, K., Matsuda, S., and Takahara, A., "Toward developing a violin playing robot-bowing by anthropomorphic robot arm and sound analysis," in *Robot and Human interactive Communication, 2007. RO-MAN 2007. The 16th IEEE International Symposium on*, pp. 763–768, IEEE, 2007.

[178] Shmulevich, I. and Povel, D.-J., "Rhythm complexity measures for music pattern recognition," in *Multimedia Signal Processing, 1998 IEEE Second Workshop on*, pp. 167–172, IEEE, 1998.

[179] Shmulevich, I. and Povel, D.-J., "Measures of temporal pattern complexity," *Journal of New Music Research*, vol. 29, no. 1, pp. 61–69, 2000.

[180] Shmulevich, I. and Povel, D.-J., "Complexity measures of musical rhythms,"

[181] Simon, I., Morris, D., and Basu, S., "Mysong: automatic accompaniment generation for vocal melodies," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 725–734, ACM, 2008.

[182] Singer, E., Feddersen, J., Redmon, C., and Bowen, B., "Lemur's musical robots," in *Proceedings of the 2004 conference on New interfaces for musical expression*, pp. 181–184, National University of Singapore, 2004.

[183] Singer, E., Larke, K., and Bianciardi, D., "Lemur guitarbot: Midi robotic string instrument," in *Proceedings of the 2003 conference on New interfaces for musical expression*, pp. 188–191, National University of Singapore, 2003.

[184] Sisbot, E. A., Marin-Urias, L. F., Alami, R., and Simeon, T., "A human aware mobile robot motion planner," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 874–883, 2007.

[185] Solis, J., Chida, K., Isoda, S., Suefuji, K., Arino, C., and Takanishi, A., "The anthropomorphic flutist robot wf-4r: from mechanical to perceptual improvements," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pp. 64–69, IEEE, 2005.

[186] Solis, J., Petersen, K., Ninomiya, T., Takeuchi, M., and Takanishi, A., "Development of anthropomorphic musical performance robots: From understanding the nature of music performance to its application to entertainment robotics," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 2309–2314, IEEE, 2009.

[187] Solis, J., Suefuji, K., Taniguchi, K., Ninomiya, T., Maeda, M., and Takanishi, A., "Implementation of expressive performance rules on the wf-4riii by modeling a professional flutist performance using nn," in *Robotics and Automation, 2007 IEEE International Conference on*, pp. 2552–2557, IEEE, 2007.

[188] Solis, J., Takanishi, A., and Hashimoto, K., "Development of an anthropomorphic saxophone-playing robot," in *Brain, Body and Machine*, pp. 175–186, Springer, 2010.

[189] Stevens, C., Lees, N., Vonwiller, J., and Burnham, D., "On-line experimental methods to evaluate text-to-speech (tts) synthesis: effects of voice gender and signal quality on intelligibility, naturalness and preference," *Computer speech & language*, vol. 19, no. 2, pp. 129–146, 2005.

[190] Stocco, L., Salcudean, S., and Sassani, F., "Fast constrained global minimax optimization of robot parameters," *Robotica*, vol. 16, no. 06, pp. 595–605, 1998.

[191] Stone, G. L., *Stick control: for the snare drummer*. Alfred Music, 2013.

[192] Tarn, T., Bejczy, A., and Yun, X., "Design of dynamic control of two cooperating robot arms: Closed chain formulation," in *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, vol. 4, pp. 7–13, IEEE, 1987.

[193] Thompson, M. and Luck, G., "Effect of pianists expressive intention on amount and type of body movement," in *10th International Conference on Music Perception and Cognition, Sapporo, Japan*, 2008.

[194] Tindale, A. R., Kapur, A., Tzanetakis, G., and Fujinaga, I., "Retrieval of percussion gestures using timbre classification techniques.," in *ISMIR*, 2004.

[195] Toiviainen, P., Luck, G., and Thompson, M. R., "Embodied meter: hierarchical eigenmodes in music-induced movement," 2010.

[196] Toussaint, G. T. and others, "A mathematical analysis of african, brazilian, and cuban clave rhythms," in *Proceedings of BRIDGES: Mathematical Connections in Art, Music and Science*, pp. 157–168, Citeseer, 2002.

[197] Toyota, "Partner robot." `http://www.toyota-global.com/innovation/partner_robot/`. Accessed:5-1-2014.

[198] Trafton, G., Hiatt, L., Harrison, A., Tamborello, F., Khemlani, S., and Schultz, A., "Act-r/e: An embodied cognitive architecture for human-robot interaction," *Journal of Human-Robot Interaction*, vol. 2, no. 1, pp. 30–55, 2013.

[199] Trimpin, "Portfolio," in *Seattle, Washington*.

[200] van den Oord, A., Kalchbrenner, N., Vinyals, O., Espeholt, L., Graves, A., and Kavukcuoglu, K., "Conditional image generation with pixelcnn decoders," *CoRR*, vol. abs/1606.05328, 2016.

[201] Vines, B. W., Krumhansl, C. L., Wanderley, M. M., and Levitin, D. J., "Cross-modal interactions in the perception of musical performance," *Cognition*, vol. 101, no. 1, pp. 80–113, 2006.

[202] Walters, M. L., Syrdal, D. S., Dautenhahn, K., Te Boekhorst, R., and Koay, K. L., "Avoiding the uncanny valley: robot appearance, personality and consistency of behavior in an attention-seeking home scenario for a robot companion," *Autonomous Robots*, vol. 24, no. 2, pp. 159–178, 2008.

[203] Wanderley, M. M., Vines, B. W., Middleton, N., McKay, C., and Hatch, W., "The musical significance of clarinetists' ancillary gestures: an exploration of the field," *Journal of New Music Research*, vol. 34, no. 1, pp. 97–113, 2005.

[204] Wang, C.-i. and Dubnov, S., "Guided music synthesis with variable markov oracle," in *3rd International Workshop on Musical Metacreation, Raleigh, NC, USA*, 2014.

[205] Wasserman, E. A. and Miller, R. R., "What's elementary about associative learning?," *Annual review of psychology*, vol. 48, no. 1, pp. 573–607, 1997.

[206] Weinberg, G., *Can robots be creative? - Gil Weinberg*. Ted-Ed, 2015.

[207] Weinberg, G., Beck, A., and Godfrey, M., "Zoozbeat: a gesture-based mobile music studio," in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, 2009.

[208] Weinberg, G., Blosser, B., Mallikarjuna, T., and Raman, A., "The creation of a multi-human, multi-robot interactive jam session," in *Proceedings of the Ninth International Conference on New Interfaces for Musical Expression*, pp. 70–73, 2009.

[209] Weinberg, G. and Driscoll, S., "Robot-human interaction with an anthropomorphic percussionist," in *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pp. 1229–1232, ACM, 2006.

[210] Weinberg, G. and Driscoll, S., "Toward robotic musicianship," *Computer Music Journal*, vol. 30, no. 4, pp. 28–45, 2006.

[211] Weinberg, G. and Driscoll, S., "The design of a perceptual and improvisational robotic marimba player," in *Robot and Human interactive Communication, 2007. RO-MAN 2007. The 16th IEEE International Symposium on*, pp. 769–774, IEEE, 2007.

[212] Weinberg, G., Driscoll, S., and Thatcher, T., "Jam'aa-a middle eastern percussion ensemble for human and robotic players," in *International Computer Music Conference*, pp. 464–467, 2006.

[213] Weinberg, G., Godfrey, M., Rae, A., and Rhoads, J., "A real-time genetic algorithm in human-robot musical improvisation," in *Computer music modeling and retrieval. Sense of sounds*, pp. 351–359, Springer, 2008.

[214] Weinberg, G., Raman, A., and Mallikarjuna, T., "Interactive jamming with shimon: a social robotic musician," in *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, pp. 233–234, ACM, 2009.

[215] Weng, J., "Developmental robotics: Theory and experiments," *International Journal of Humanoid Robotics*, vol. 1, no. 02, pp. 199–236, 2004.

[216] Whalley, I., "Generative improv. & interactive music project (giimp)," *Proceedings of NIME 2010*, pp. 255–258, 2010.

[217] Williamson, M. M., "Rhythmic robot arm control using oscillators," in *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, vol. 1, pp. 77–83, IEEE, 1998.

[218] Williamson, M. M., *Robot arm control exploiting natural dynamics*. PhD thesis, Massachusetts Institute of Technology, 1999.

[219] Wilson, M., "Six views of embodied cognition," *Psychonomic bulletin & review*, vol. 9, no. 4, pp. 625–636, 2002.

[220] Wilson, R. A. and Foglia, L., "Embodied cognition," 2011.

[221] Wright, M., "Open sound control: an enabling technology for musical networking," *Organised Sound*, vol. 10, no. 3, p. 193, 2005.

[222] Xu, L., Zhang, D., Wang, K., and Wang, L., "Arrhythmic pulses detection using lempel-ziv complexity analysis," *EURASIP Journal on Advances in Signal Processing*, vol. 2006, no. 1, pp. 1–12, 2006.

[223] Zen, H., Tokuda, K., and Black, A. W., "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[224] Zhang, A., Malhotra, M., and Matsuoka, Y., "Musical piano performance by the act hand," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 3536–3541, IEEE, 2011.

[225] Ziemke, T., "Cybernetics and embodied cognition: on the construction of realities in organisms and robots," *Kybernetes*, vol. 34, no. 1/2, pp. 118–128, 2005.